

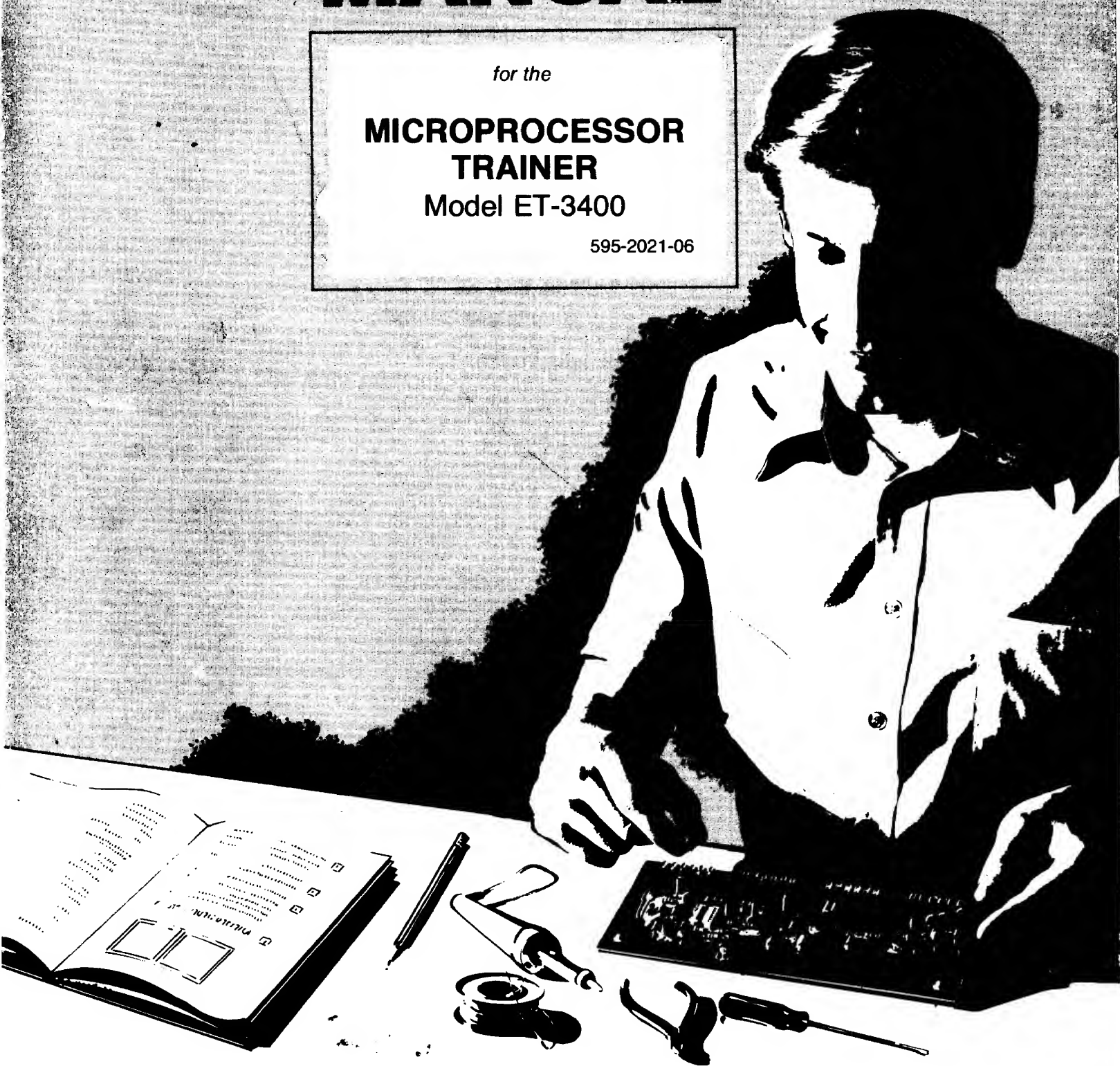
HEATHKIT[®] MANUAL

for the

MICROPROCESSOR TRAINER

Model ET-3400

595-2021-06



HEATH COMPANY • BENTON HARBOR, MICHIGAN

HEATH COMPANY PHONE DIRECTORY

The following telephone numbers are direct lines to the departments listed:

Kit orders and delivery information (616) 982-3411
Credit (616) 982-3561
Replacement Parts (616) 982-3571

Technical Assistance Phone Numbers

8:00 A.M. to 12 P.M. and 1:00 P.M. to 4:30 P.M., EST, Weekdays Only
R/C, Audio, and Electronic Organs (616) 982-3310
Amateur Radio (616) 982-3296
Test Equipment, Weather Instruments and
Home Clocks (616) 982-3315
Television (616) 982-3307
Aircraft, Marine, Security, Scanners, Automotive,
Appliances and General Products (616) 982-3496
Computers (616) 982-3309

YOUR HEATHKIT 90-DAY LIMITED WARRANTY

For a period of ninety (90) days after purchase, Heath Company will replace or repair free of charge any parts that are defective either in materials or workmanship. You can obtain parts directly from Heath Company by writing us at the address below or by telephoning us at (616) 982-3571. And we'll pay shipping charges to get those parts to you — anywhere in the world.

We warrant that during the first ninety (90) days after purchase, our products, when correctly assembled, calibrated, adjusted and used in accordance with our printed instructions, will meet published specifications.

If a defective part or error in design has caused your Heathkit product to malfunction during the warranty period through no fault of yours, we will service it free upon proof of purchase and delivery at your expense to the Heath factory, any Heathkit Electronic Center, or any of our authorized overseas distributors.

You will receive free consultation on any problem you might encounter in the assembly or use of your Heathkit product. Just drop us a line or give us a call. Sorry, we cannot accept collect calls.

Our warranty does not cover and we are not responsible for damage caused by: incorrect assembly, the use of corrosive solder, defective tools, misuse, or fire; or by unauthorized modifications to or uses of our products for purposes other than as advertised. Our warranty does not include reimbursement for inconvenience, loss of use, customer assembly or set-up time.

This warranty covers only Heathkit products and is not extended to allied equipment or components used in conjunction with our products. **We are not responsible for accidental or consequential damages.** Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

If you are not satisfied with our service (warranty or otherwise) or with our products, write directly to our Director of Customer Services, Heath Company, Benton Harbor, Michigan 49022. He will make certain your problems receive immediate, personal attention.

HEATH COMPANY
BENTON HARBOR, MI. 49022

The Heath Company reserves the right to discontinue products and to change specifications at any time without incurring any obligation to incorporate new features in products previously sold.

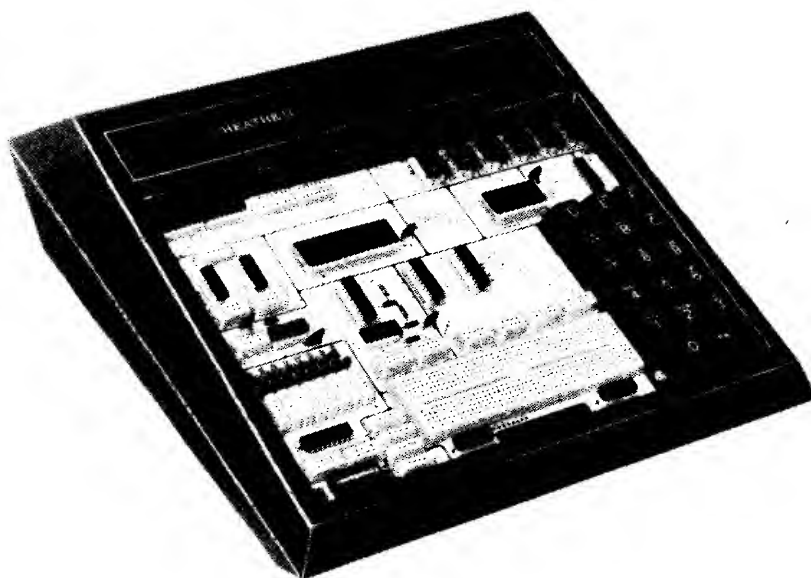
Heathkit® Manual

for the

MICROPROCESSOR TRAINER

Model ET-3400

595-2021-06



HEATH COMPANY
BENTON HARBOR, MICHIGAN 49022

Copyright © 1977
Heath Company
All Rights Reserved
Printed in the United States of America



TABLE OF CONTENTS

Introduction	3	In Case of Difficulty	91
Assembly Notes	4	Troubleshooting Charts	93
Parts List	7	Specifications	107
Step-by-Step Assembly		Theory of Operation	108
Main Circuit Board	10	Semiconductor Identification Charts	110
Keyboard Circuit Board	24	Circuit Board	
Support Bracket Assembly	27	X-Ray View	(Illustration Booklet, Page 13)
Cabinet Assembly and Wiring	30	Schematic	Fold-in
Initial Tests		Warranty	Inside front cover
Voltage Tests	34	Customer Service	Inside rear cover
Tests Continued	35		
Operational Tests	37		
Final Assembly	42		
Operation	45		



INTRODUCTION

The ET-3400 Microcomputer Learning System is a practical, low cost microprocessor trainer; designed as a learning tool to teach microprocessor operation, programming, and applications. The ET-3400 Trainer is designed to accompany the EE-3401 Individual Learning Program on microprocessors. All of the programming and hardware interface experiments supplied with this course are implemented on the Trainer. While the Trainer was designed primarily to accompany this course, it is a flexible, general-purpose training unit and microprocessor breadboard. It can be used in many other applications that require a low cost microprocessor-based software development system or as a design aid for developing special interfaces.

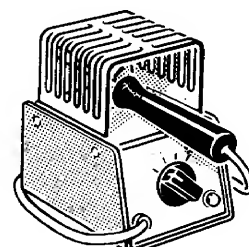
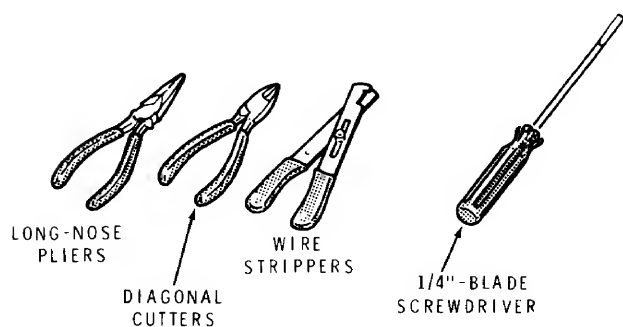
MAIN FEATURES

- Uses the popular 6800 Microprocessor.
- Is supplied with 256 bytes of semiconductor RAM (expandable to 512 bytes).
- Features 1K ROM monitor program.
- Has hexadecimal keyboard for rapid data and program entry.
- Has six digits of hexadecimal display for reading out memory addresses, their contents, and register contents.
- Uses breadboarding sockets that permit rapid, solderless assembly of IC logic circuitry to be used with the microprocessor. They are ideal for prototyping special interface circuits.
- The microprocessor address bus, data bus, control lines, and associated signals are buffered and terminated on front panel connectors; allowing complete freedom in experimenting with the microprocessor and its associated circuitry.
- Has eight individual, independent, binary LED indicators for monitoring logic states in the breadboard circuitry.
- Has eight individual, independent, binary data switches that can be used for supplying binary words and logic levels in the breadboarding circuitry.
- The built-in power supplies furnish power to all internal circuitry and have sufficient reserve to power breadboard circuits. The +5 and ± 12 -volt supply voltages are connected to front panel connectors.
- Has provision for future expansion of memory and I/O capabilities.

ASSEMBLY NOTES

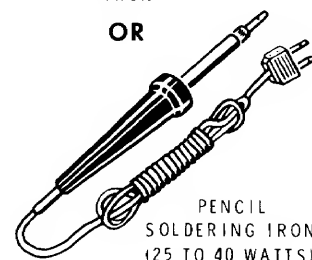
TOOLS

You will need these tools to assemble your kit.

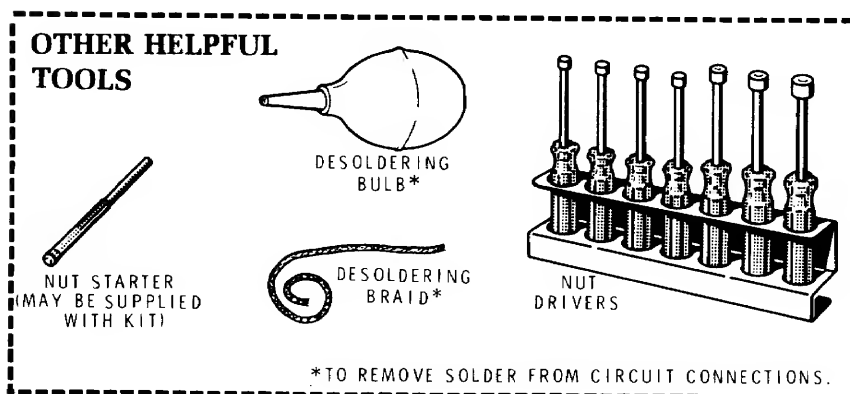


HEATHKIT
SOLDERING
IRON

OR



PENCIL
SOLDERING IRON
(25 TO 40 WATTS)



ASSEMBLY

1. Follow the instructions carefully. Read the entire step before you perform each operation.
2. The illustrations in the Manual are called Pictorials and Details. Pictorials show the overall operation for a group of assembly steps; Details generally illustrate a single step. When you are directed to refer to a certain Pictorial "for the following steps," continue using that Pictorial until you are referred to another Pictorial for another group of steps.
3. Most kits use a separate "Illustration Booklet" that contains illustrations (Pictorials, Details, etc.) that are too large for the Assembly Manual. Keep the "Illustration Booklet" with the Assembly Manual. The illustrations in it are arranged in Pictorial number sequence.
4. Position all parts as shown in the Pictorials.
5. Solder a part or a group of parts only when you are instructed to do so.



6. Each circuit part in an electronic kit has its own component number (R2, C4, etc.). Use these numbers when you want to identify the same part in the various sections of the Manual. These numbers, which are especially useful if a part has to be replaced, appear:
 - In the Parts List,
 - At the beginning of each step where a component is installed,
 - In some illustrations,
 - In the Schematic,
 - In the section at the rear of the Manual.
7. When you are instructed to cut something to a particular length, use the scales (rulers) provided at the bottom of the Manual pages.

SAFETY WARNING: Avoid eye injury when you cut off excess lead lengths. Hold the leads so they cannot fly toward your eyes.

SOLDERING

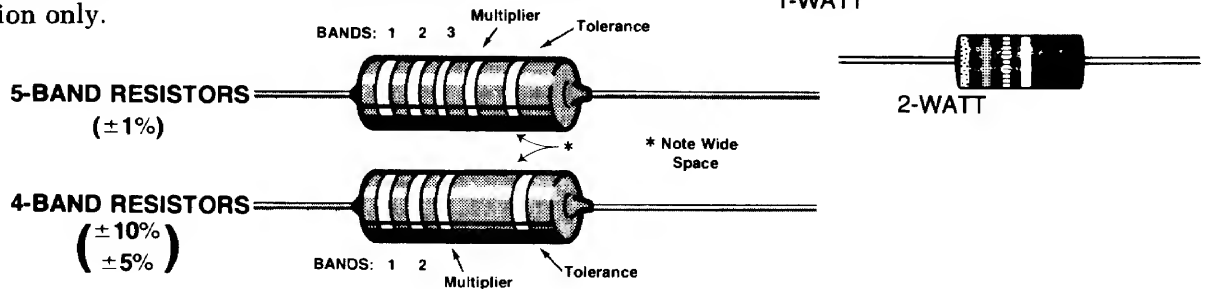
Soldering is one of the most important operations you will perform while assembling your kit. A good solder connection will form an electrical connection between two parts, such as a component lead and a circuit board foil. A bad solder connection could prevent an otherwise well-assembled kit from operating properly.

It is easy to make a good solder connection if you follow a few simple rules:

1. Use the right type of soldering iron. A 25 to 40-watt pencil soldering iron with a 1/8" or 3/16" chisel or pyramid tip works best.
2. Keep the soldering iron tip clean. Wipe it often on a wet sponge or cloth; then apply solder to the tip to give the entire tip a wet look. This process is called tinning, and it will protect the tip and enable you to make good connections. When solder tends to "ball" or does not stick to the tip, the tip needs to be cleaned and retinned.

PARTS

Resistors will be called out by their resistance value in Ω (ohms), $k\Omega$ (kilohms), or $M\Omega$ (megohms). Certain types of resistors will have the value printed on the body, while others will be identified by a color code. The colors of the bands and the value will be given in the steps, therefore the following color code is given for information only.



Band 1 1st Digit		Band 2 2nd Digit		Band 3 (if used) 3rd Digit		Multiplier		Resistance Tolerance	
Color	Digit	Color	Digit	Color	Digit	Color	Multiplier	Color	Tolerance
Black	0	Black	0	Black	0	Black	1	Silver	$\pm 10\%$
Brown	1	Brown	1	Brown	1	Brown	10	Gold	$\pm 5\%$
Red	2	Red	2	Red	2	Red	100	Brown	$\pm 1\%$
Orange	3	Orange	3	Orange	3	Orange	1,000		
Yellow	4	Yellow	4	Yellow	4	Yellow	10,000		
Green	5	Green	5	Green	5	Green	100,000		
Blue	6	Blue	6	Blue	6	Blue	1,000,000		
Violet	7	Violet	7	Violet	7	Silver	0.01		
Gray	8	Gray	8	Gray	8	Gold	0.1		
White	9	White	9	White	9				

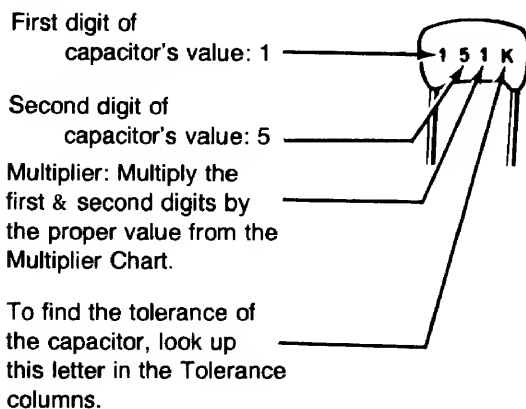
Capacitors will be called out by their capacitance value in μF (microfarads) or pF (picofarads) and type: ceramic, Mylar*, electrolytic, etc. Some capacitors may have their value printed in the following manner:

EXAMPLES:

$$151K = 15 \times 10 = 150 \text{ pF}$$

$$759 = 75 \times 0.1 = 7.5 \text{ pF}$$

NOTE: The letter "R" may be used at times to signify a decimal point: as in: 2R2 = 2.2 (pF or μF).



MULTIPLIER		TOLERANCE OF CAPACITOR		
FOR THE NUMBER:	MULTIPLY BY:	10 pF OR LESS	LETTER	OVER 10 pF
0	1	$\pm 0.1 \text{ pF}$	B	
1	10	$\pm 0.25 \text{ pF}$	C	
2	100	$\pm 0.5 \text{ pF}$	D	
3	1000	$\pm 1.0 \text{ pF}$	F	$\pm 1\%$
4	10,000	$\pm 2.0 \text{ pF}$	G	$\pm 2\%$
5	100,000		H	$\pm 3\%$
			J	$\pm 5\%$
8	0.01		K	$\pm 10\%$
9	0.1		M	$\pm 20\%$



PARTS LIST

Check each part against the following list. Any part that is packed in an individual envelope with the part number on it should be placed back in the envelope after you identify it until it is called for in a step. Do not discard any packing materials until all parts are accounted for.

The key numbers correspond to the numbers on the "Parts Pictorial" in the separate "Illustration Booklet" on Pages 1 and 2.

To order a replacement part: Always include the PART NUMBER. Use the Parts Order Form furnished with the kit. If one is not available, see "Replacement Parts" inside the rear cover of the Manual. Your Warranty is located inside the front cover. For prices, refer to the separate "Heath Parts Price List."

KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.
------------	-------------------	------	-------------	----------------------

RESISTORS

NOTES:

1. All resistors are 10% tolerance unless otherwise noted. A fourth color band of silver indicates a 10% tolerance; a fourth band of gold indicates 5% tolerance.
2. The resistors may be packed in more than one envelope. Open all the resistor envelopes in this pack before you check them against the Parts List.

1/4-Watt Resistors

A1	6-151-12	1	150 Ω (brown-green-brown)	R2
A1	6-181-12	9	180 Ω , 5% (brown-gray-brown)	R32 through R39, R107
A1	6-471-12	48	470 Ω , 5% (yellow-violet-brown)	R58 through R105
A1	6-122-12	1	1200 Ω , 5% (brown-red-red)	R49
A1	6-472-12	9	4700 Ω (yellow-violet-red)	R16 through R23, R106
A1	6-682-12	1	6800 Ω (blue-gray-red)	R6
A1	6-822-12	26	8200 Ω (gray-red-red)	R5, R10, R15, R24 through R31, R40, R41, R43 through R48, R51 through R57

KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.
------------	-------------------	------	-------------	----------------------

Resistors (cont'd.)

A1	6-153-12	1	15 k Ω (brown-green-orange)	R8
A1	6-273-12	2	27 k Ω (red-violet-orange)	R1, R42
A1	6-104-12	3	100 k Ω (brown-black-yellow)	R11, R12, R14
A1	6-154-12	1	150 k Ω (brown-green-yellow)	R9
A1	6-224-12	2	220 k Ω (red-red-yellow)	R7, R50
A1	6-824-12	1	820 k Ω (gray-red-yellow)	R13

Other Resistors

A2	6-680	2	68 Ω , 1/2-watt (blue-gray-black)	R3, R4
----	-------	---	--	--------

CAPACITORS

Electrolytic Capacitors

B1	25-200	1	.68 μ F tantalum	C13
B1	25-221	2	2.2 μ F tantalum	C8, C9
B1	25-220	2	10 μ F tantalum (10M)	C11, C12
B2	25-241	2	1200 μ F	C6, C7
B2	25-272	1	6000 μ F	C1

Other Capacitors

B3	20-102	1	100 pF mica	C23
B4	21-176	12	.01 μ F ceramic	C4, C5, C14 through C22, C24
B5	27-85	2	.22 μ F Mylar	C2, C3



KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.
DIODES				
C1	56-56	4	1N4149 diode	D7 through D10
C1	57-42	2	3A1 diode	D1, D2
C1	57-65	4	1N4002 diode	D3, D4, D5, D6

INTEGRATED CIRCUITS (IC's)

NOTES:

- Integrated circuits are marked for identification in one of the following four ways:
 - Part number.
 - Type number. (For integrated circuits, this refers only to the numbers; the letters may be different or missing.)
 - Part number and type number.
 - Part number with a type number other than the one listed.
- Some of the IC's may be packed in conductive foam. Do not remove the IC's from the foam until you are instructed to do so.

D1	442-30	1	μ A309K	IC31
D2	442-644	1	LM78L12	IC29
D2	442-646	1	MC79L12AC	IC30
D3	442-616	1	LM3302N, LM2901N or μ A775	IC18
D3	443-717	1	74126N	IC4
D3	443-26	2	74S00	IC5, IC21
D3	443-839	2	74LS243	IC9, IC10
D4	443-720	1	40097	IC13
D4	443-721	2	2112-2	IC14 through IC17
D4	443-804	6	74LS259	IC23 through IC28
D4	443-807	4	74LS42	IC2, IC3, IC20, IC22
D4	443-840	1	MC6875	IC19
D5	443-824	4	74LS241	IC1, IC6, IC7, IC8
D6	444-17	1	MCM6830A	IC12
D7	443-827	1	MC6800P	IC11

KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.
SWITCHES — INSULATORS				
E1	60-34	1	Rocker switch	SW1
E2	60-621	1	Switch assembly (May be slide or rocker switches.)	
E3	64-839	17	Pushbutton switch (May look different than one shown.)	
E4	73-4	1	Rubber grommet	
E5	75-724	1	Insulator plate	
E6	75-788	2	Insulating paper	

HARDWARE

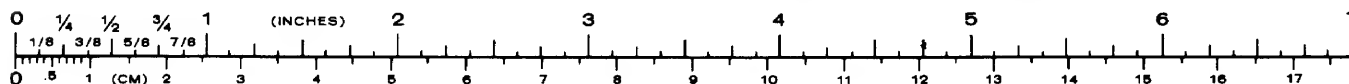
NOTE: The hardware may be in more than one packet. Open all the hardware packets according to their size before you check the hardware.

Hardware is shown actual size. To identify a piece of hardware, place it over the illustration.

F1	250-163	3	#4 \times 5/16" self-tapping screw
F2	250-138	2	6-32 \times 3/16" screw
F3	250-56	13	6-32 \times 1/4" screw
F4	250-475	10	#6 \times 3/8" hex head screw
F5	250-32	1	6-32 \times 3/8" flat head screw
F6	250-162	2	6-32 \times 1/2" screw
F7	250-559	8	#6 \times 5/8" self-tapping screw
F8	250-1137	2	#6 \times 1-1/8" self-tapping screw
F9	252-3	6	6-32 nut
F10	254-1	11	#6 lockwasher
F11	255-23	4	Spacer
F12	259-1	3	#6 solder lug
F13	259-22	1	Spade lug
F14	260-56	2	Fuse clip

WIRE — BRAID — LINE CORD

344-51	18"	Brown wire
344-52	3"	Red wire
344-53	9"	Orange wire
344-54	20'	Yellow wire
344-71	18"	White-brown wire
344-74	9"	White-yellow wire
344-73	9"	White-orange wire
344-99	18"	White stranded wire
345-1	3"	Flat braid
89-49	1	Line cord
346-1	2"	Sleeving





KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.
---------	----------------	------	-------------	-------------------

TERMINAL STRIPS — CONNECTORS — SOCKETS

G1	431-2	1	2-lug terminal strip	
G2	431-86	1	6-lug terminal strip	
G3	432-874	13	4-pin connector block	
G4	432-973	11	8-pin connector block	
G5	432-875	1	Large connector block	
G6	432-921	2	3-pin IC socket	
G7	434-336	1	TO-3 socket	
G8	434-298	12	14-pin IC socket	
G9	434-299	16	16-pin IC socket	
G10	434-311	4	20-pin IC socket	
G11	434-307	1	24-pin IC socket	
G12	434-253	1	40-pin IC socket	

CIRCUIT BOARDS — CABINET — BRACKET

	85-2033-3	1	Main circuit board	
	85-2010-1	1	Keyboard circuit board	
H1	92-611	1	Cabinet top	
H2	92-612	1	Cabinet bottom	
H3	204-2291	1	Support bracket	

LIGHT-EMITTING DIODES (LED's) — FUSE

J1	411-831	6	7-segment LED	H, I, N, Z, V, C
J2	412-640	1	3/8" red LED	LED1

KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.
---------	----------------	------	-------------	-------------------

Light-Emitting Diodes (Led's) — Fuse (cont'd.)

J3	412-616	8	1/4" red LED	LED2 through LED9
J4	421-42	1	3/8-ampere, 3AG, slow-blow fuse	F1

MISCELLANEOUS

K1	54-920	1	Power transformer	T1
K2	260-700	1	LED grommet	
K3	261-34	4	Foot	
K4	352-13	1	Silicone grease	
K5	354-7	1	Cable tie	
K6	262-8	2	Terminal pin	
K7	475-12	1	Ferrite bead	
K8	462-1023	7	Square knob	
K9	490-111	1	IC puller	
			Solder	

PRINTED MATERIAL

L1	390-1255	1	Fuse label	
L2	390-1390	1	Power label	
L3	390-1391	1	"Heathkit" label	
L4	390-1395	1	Keyboard label set	
	390-1404	1	Red label set	
L5		1	Blue and white label	
	597-260	1	Parts Order Form	
		1	Assembly Manual (See Page 1 for part number.)	

STEP-BY-STEP ASSEMBLY

The steps performed in this Pictorial are in this area of the circuit board.

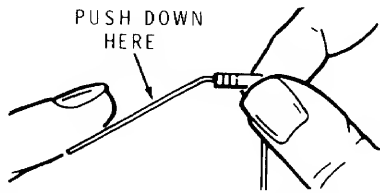
MAIN CIRCUIT BOARD

START

In the following steps you will be given detailed instructions on how to install and solder the first part on the circuit board. Read and perform each step carefully. Then use the same procedure whenever you install parts on a circuit board.

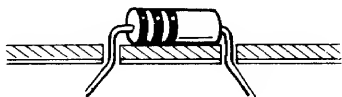
- () Position the circuit board as shown in the identification drawing with the printed side up.

- () R49: Hold a 1200 Ω (brown-red-red) resistor by the body as shown and bend the leads straight down.

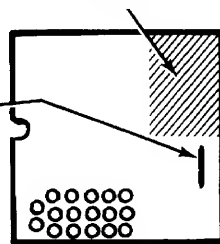


- () Push the leads through the holes at the indicated location on the circuit board. The end with color bands may be positioned either way.

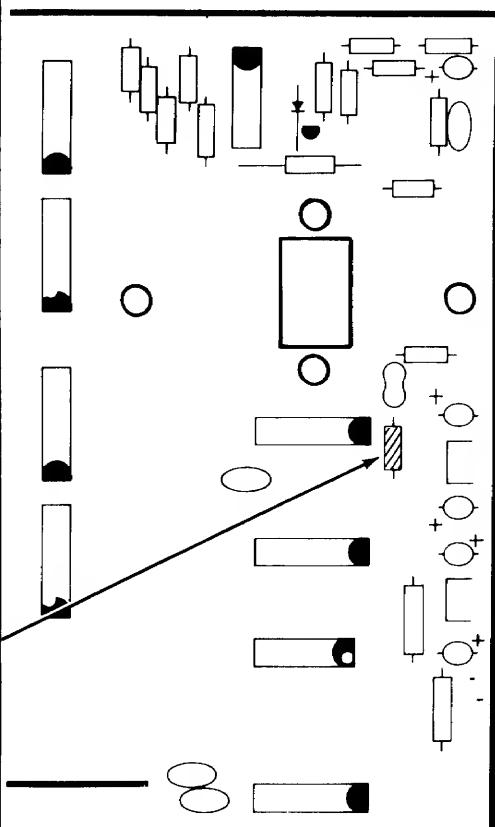
- () Press the resistor against the circuit board. Then bend the leads outward slightly to hold the resistor in place.



PART
NUMBER



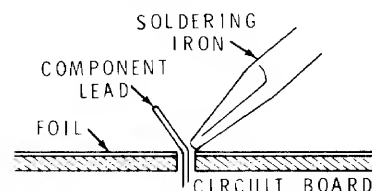
IDENTIFICATION
DRAWING



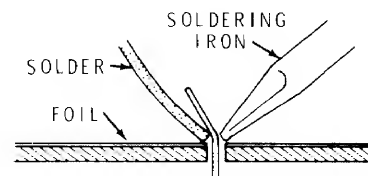
CONTINUE

- () Solder the resistor leads to the circuit board as follows:

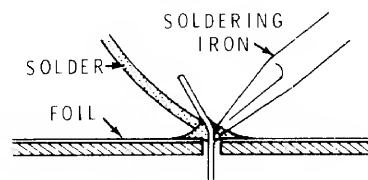
1. Push the soldering iron tip against both the lead and the circuit board foil. Heat both for two or three seconds.



2. Then apply solder to the other side of the connection. IMPORTANT: Let the heated lead and the circuit board foil melt the solder.



3. As the solder begins to melt, allow it to flow around the connection. Then remove the solder and the iron and let the connection cool.

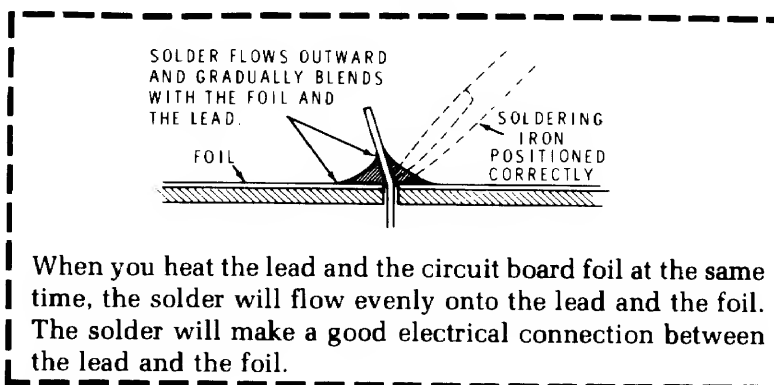


- () Hold the lead with one hand while you cut off the excess lead length close to the connection. This will keep you from being hit in the eye by the flying lead.

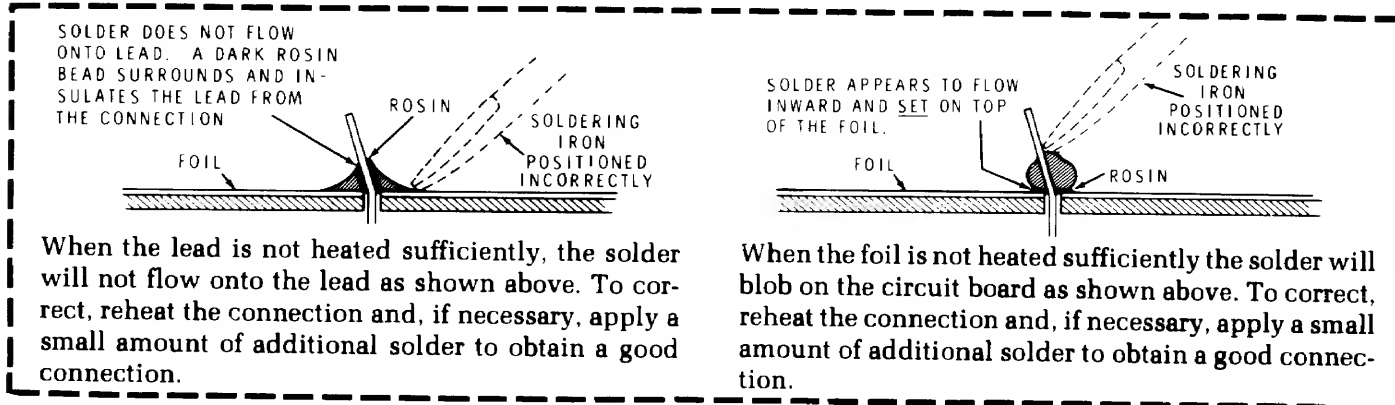
- () Check the connection. Compare it to the illustrations on the next page. After you have checked the solder connections, proceed with the assembly on page 12. Use the same soldering procedure for each connection.

PICTORIAL 1-1

A GOOD SOLDER CONNECTION



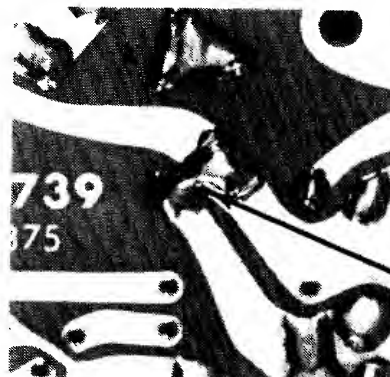
POOR SOLDER CONNECTIONS



SOLDER BRIDGES

A solder bridge between two adjacent foils is shown in photograph A. Photograph B shows how the connection should appear. A solder bridge may occur if you accidentally touch an adjacent previously soldered connection, if you use too much solder, or if you "drag" the soldering iron across other foils as you remove it from the connection. A good rule to follow is: always take a good look at the foil area around each lead before you solder it. Then, when you solder the connection, make sure the solder remains in this area and does not bridge to another foil. This is especially important when the foils are small and close together. NOTE: It is alright for solder to bridge two connections on the same foil.

Use only enough solder to make a good connection, and lift the soldering iron straight up from the circuit board. If a solder bridge should develop, turn the circuit board foil-side-down and heat the solder between connections. The excess solder will run onto the tip of the soldering iron, and this will remove the solder bridge. NOTE: The foil side of most circuit boards has a coating on it called "solder resist." This is a protective insulation to help prevent solder bridges.



A

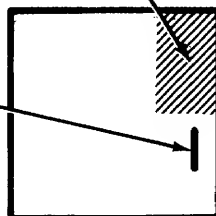
SOLDER BRIDGE

B



The steps performed in this Pictorial are in this area of the circuit board.

PART
NUMBER



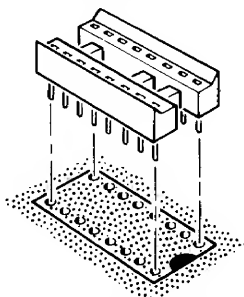
IDENTIFICATION
DRAWING

MAIN CIRCUIT BOARD

START

NOTE: When you install an IC socket, use the following procedure:

1. Be sure all the pins are straight.
2. Insert the pins into the holes.
3. Turn the circuit board over and be sure the correct number of pins extend from the board. If not, one or more pins may be bent under the socket. Remove the socket, straighten the pins, and reinstall the socket.
4. Solder the pins to the foil as you install each socket.
NOTE: Some socket pins will have no foil pads; do not solder these pins.



16-pin IC sockets at the seven following locations:

- () IC14. () IC19.
() IC15. () IC20.
() IC16. () IC22.
() IC17.

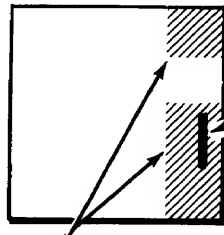
NOTE: Be sure you install the first resistor (Page 10).

CONTINUE

- () R10: 8200 Ω (gray-red-red).
() R7: 220 k Ω (red-red-yellow).
() R9: 150 k Ω (brown-green-yellow).
() R2: 150 Ω (brown-green-brown).
() R14: 100 k Ω (brown-black-yellow).
() R12: 100 k Ω (brown-black-yellow).
() R13: 820 k Ω (gray-red-yellow).
() R11: 100 k Ω (brown-black-yellow).
() R15: 8200 Ω (gray-red-red).
() R5: 8200 Ω (gray-red-red).
() R6: 6800 Ω (blue-gray-red).
() 14-pin IC socket at IC18.
() R50: 220 k Ω (red-red-yellow).
() R1: 27 k Ω (red-violet-orange).
() R8: 15 k Ω (brown-green-orange).
() R3: 68 Ω , 1/2-watt (blue-gray-black).
() R4: 68 Ω , 1/2-watt (blue-gray-black).
() Solder the leads to the foil and cut off the excess lead lengths.
() 14-pin IC socket at IC21.

PICTORIAL 1-2

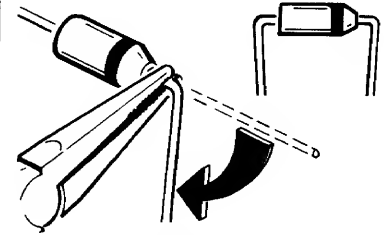
IDENTIFICATION DRAWING



PART
NUMBER

The steps performed in this Pictorial are
in these areas of the circuit board.

NOTE: Hold the leads with a pair of
long-nosed pliers close to the body of
the diode. Then bend the leads down.



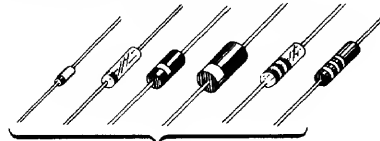
Detail 1-3A

START

() Install thirty 470 Ω (yellow-violet-brown) resistors in the area shown. After you install each group of five or six resistors, solder their leads to the foil and cut off the excess lead lengths. NOTE: See "Circuit Board X-RAY Views" in the "Illustration Booklet" for circuit component numbers.

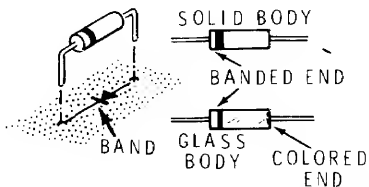
In the next column you will install diodes. Be sure you install each diode as follows.

IMPORTANT: THE BANDED END OF DIODES CAN BE MARKED IN A NUMBER OF WAYS.

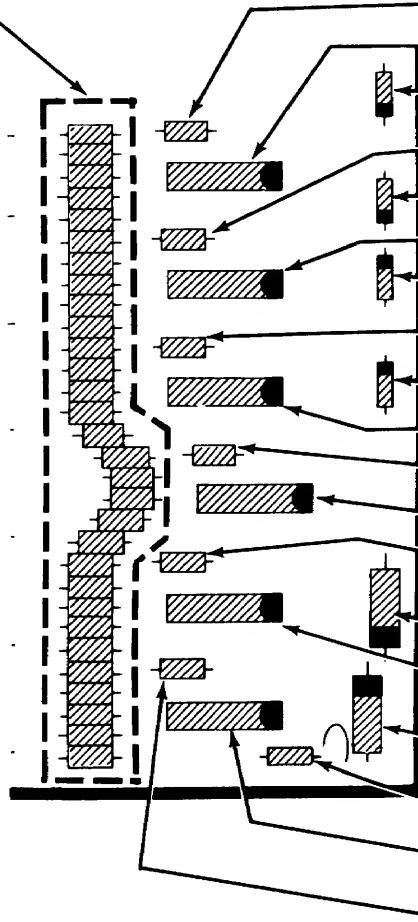
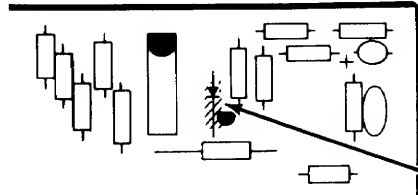


BANDED END

CAUTION: ALWAYS POSITION THE
BANDED END AS SHOWN ON THE
CIRCUIT BOARD.



If your diode has a solid body, the band is clearly defined. If your diode has a glass body, do not mistake the colored end inside the diode for the banded end. Look for a band painted on the outside of the glass.

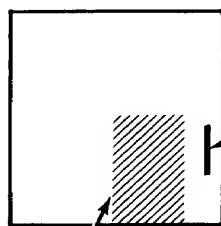


CONTINUE

NOTE: As you install the remaining
components on this Pictorial, solder
the leads to the foil and cut off the
excess lead lengths.

- () D10: 1N4149 diode (#56-56).
- () R63: 470 Ω (yellow-violet-brown).
- () 16-pin IC socket at IC23.
- () D5: 1N4002 diode (#57-65).
- () R71: 470 Ω (yellow-violet-brown).
- () D6: 1N4002 diode (#57-65).
- () 16-pin IC socket at IC24.
- () D4: 1N4002 diode (#57-65).
- () R79: 470 Ω (yellow-violet-brown).
- () D3: 1N4002 diode (#57-65).
- () 16-pin IC socket at IC25.
- () R87: 470 Ω (yellow-violet-brown).
- () 16-pin IC socket at IC26.
- () R95: 470 Ω (yellow-violet-brown).
- () D1: 3A1 diode (#57-42). See Detail 1-3A.
- () 16-pin IC socket at IC27.
- () D2: 3A1 diode (#57-42). See Detail 1-3A.
- () R106: 4700 Ω (yellow-violet-red).
- () 16-pin IC socket at IC28.
- () R104: 470 Ω (yellow-violet-brown).

PICTORIAL 1-3

IDENTIFICATION
DRAWINGPART
NUMBER

The steps performed in this Pictorial are
in this area of the circuit board.

START ➡

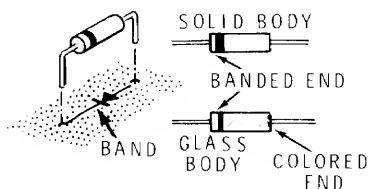
14-pin IC sockets at the six following
locations:

- () H. () Z.
() L. () V.
() N. () C.

() 24-pin IC socket at IC12.

NOTE: When you install the diodes in
the next three steps, be sure to position
the banded ends as shown.

CAUTION: ALWAYS POSITION THE
BANDED END AS SHOWN ON THE
CIRCUIT BOARD.



If your diode has a solid body, the band
is clearly defined. If your diode has a
glass body, do not mistake the colored
end inside the diode for the banded
end. Look for a band painted on the
outside of the glass.

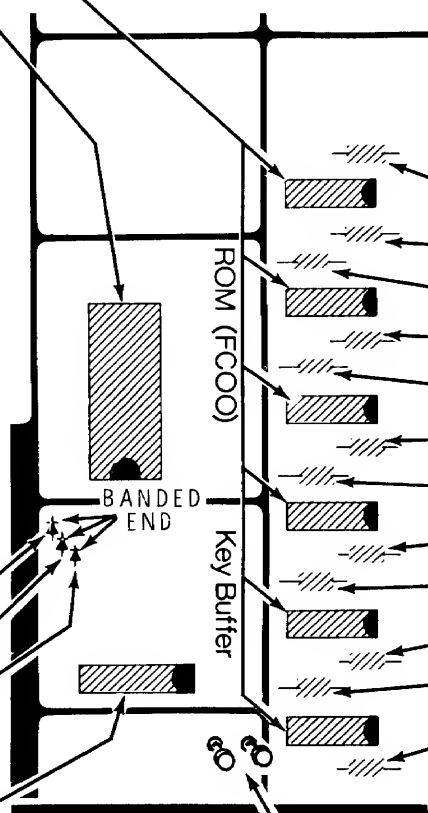
() D7: 1N4149 diode (#56-56).

() D8: 1N4149 diode (#56-56).

() D9: 1N4149 diode (#56-56).

() Solder the leads to the foil and cut
off the excess lead lengths.

() 16-pin IC socket at IC13.

**CONTINUE** ➡

Install twelve 470 Ω (yellow-violet-
brown) resistors.



() R62.

() R61.

() R70.

() R69.

() R78.

() R77.

() R86.

() R85.

() R94.

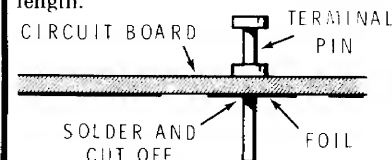
() R93.

() R102.

() R101.

() Solder the leads to the foil and cut
off the excess lead lengths.

NOTE: When you install a terminal
pin, push the pin as far as possible into
the circuit board hole. Then solder the
pin to the foil and cut off the excess pin
length.

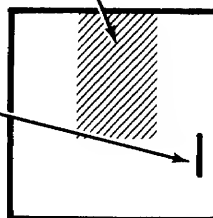


() Two terminal pins at "SEGMENT
TEST."

PICTORIAL 1-4

The steps performed in this Pictorial are in this area of the circuit board.

PART
NUMBER



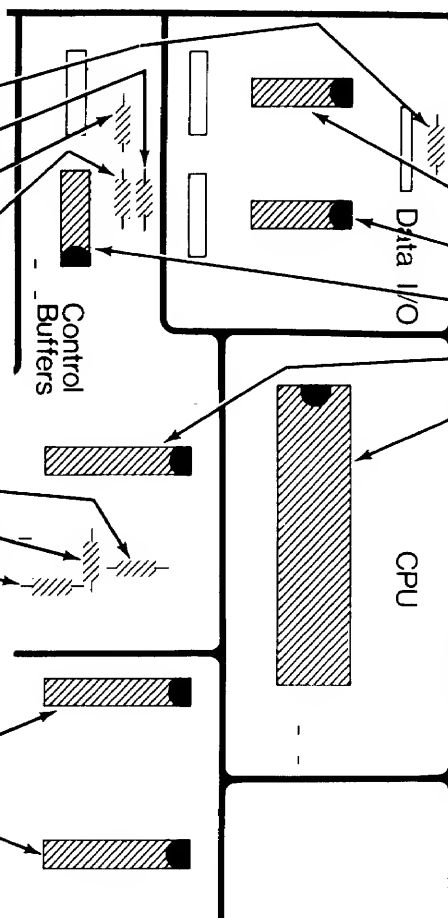
IDENTIFICATION
DRAWING

START ➡

- () R51: 8200 Ω (gray-red-red).
- () R42: 27 k Ω (red-violet-orange).
- () R43: 8200 Ω (gray-red-red).
- () R45: 8200 Ω (gray-red-red).



- () R40: 8200 Ω (gray-red-red).
- () R46: 8200 Ω (gray-red-red).
- () R48: 8200 Ω (gray-red-red).
- () Solder the leads to the foil and cut off the excess lead lengths.
- () 20-pin IC socket at IC7. Be sure all the pins protrude through the board before you solder any of them.
- () 20-pin IC socket at IC8.

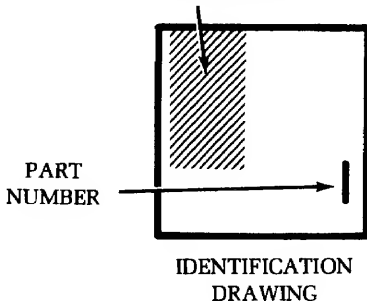


CONTINUE ➡

- () 14-pin IC socket at IC9.
- () 14-pin IC socket at IC10.
- () 14-pin IC socket at IC4.
- () 20-pin IC socket at IC6.
- () 40-pin IC socket at IC11.

PICTORIAL 1-5

The steps performed in this Pictorial are in this area of the circuit board.



START

Install four 4700 Ω (yellow-violet-red) resistors.

- () R16.
- () R17.
- () R22.
- () R23.
- () Solder the leads to the foil and cut off the excess lead lengths.
- () 20-pin IC socket at IC1.

CONTINUE

Install eight 8200 Ω (gray-red-red) resistor.

- () R24.
- () R25.
- () R26.
- () R27.
- () R28.
- () R29.
- () R30.
- () R31.
- () Solder the leads to the foil and cut off the excess lead lengths.

Install three 8200 Ω (gray-red-red) resistors.

- () R44.
- () R41.
- () R47.
- () Solder the leads to the foil and cut off the excess lead lengths.

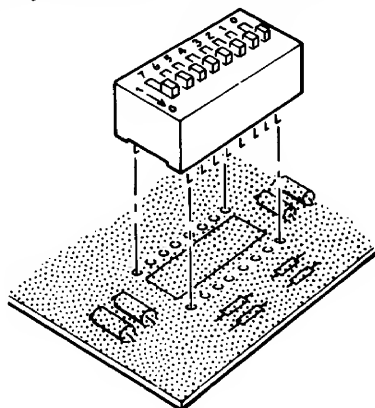
- () 16-pin IC socket at IC2.
- () 14-pin IC socket at IC5.
- () 16-pin IC socket at IC3.

- () 1" bare wire. Remove 1" of insulation from the yellow wire. Then cut off the bare wire, install it, solder its ends to the foil, and cut off the excess wire ends.

PICTORIAL 1-6

START ➡

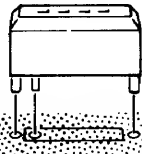
NOTE: In the next step, be sure to position the switch assembly as shown. (It may have slide or rocker switches.)



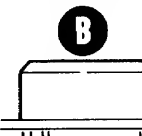
- () Switch assembly (#60-621).
- () R19: 4700 Ω (yellow-violet-red).
- () R18: 4700 Ω (yellow-violet-red).
- () R20: 4700 Ω (yellow-violet-red).
- () R21: 4700 Ω (yellow-violet-red).
- () C5: .01 μ F ceramic. See Detail 1-7A.
- () C20: .01 μ F ceramic. See Detail 1-7A.
- () Solder the leads to the foil and cut off the excess lead lengths.

Install connector blocks in the following manner:

A
4-PIN
CONNECTOR
BLOCK

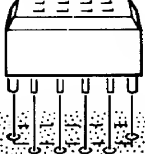


BE SURE THE METAL TAB IS STRAIGHT. THEN INSTALL THE BLOCK.



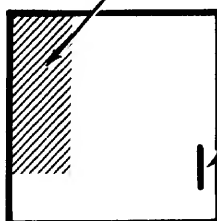
SOLDER THE METAL TAB TO THE FOIL.

**8-PIN
CONNECTOR
BLOCK**



FLATTEN THE END POSTS WITH YOUR SOLDERING IRON TIP.

The steps performed in this Pictorial are in this area of the circuit board.



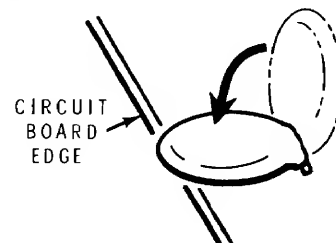
IDENTIFICATION
DRAWING

PART
NUMBER

CONTINUE ➡

- () Install two 8-pin connector blocks.
- () Install eight 4-pin connector blocks.
- () Install four 4-pin connector blocks.
- () Install six 8-pin connector blocks.

Position the capacitor down flat toward the edge of the circuit board as shown.



Detail 1-7A

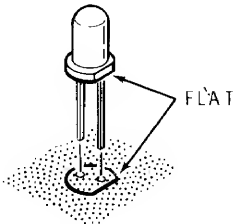
PICTORIAL 1-7

IDENTIFICATION
DRAWING

The steps performed in this Pictorial are in this area of the circuit board.

PART
NUMBER**CONTINUE** →**START** →

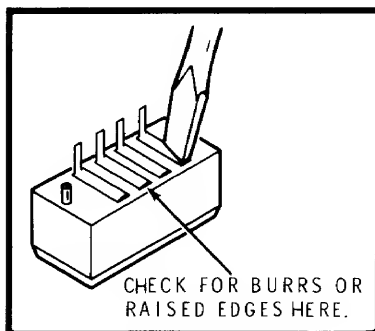
IMPORTANT: As you install LED's in the following step, be sure to match the flat on each LED with the outline of the flat on the circuit board as shown.



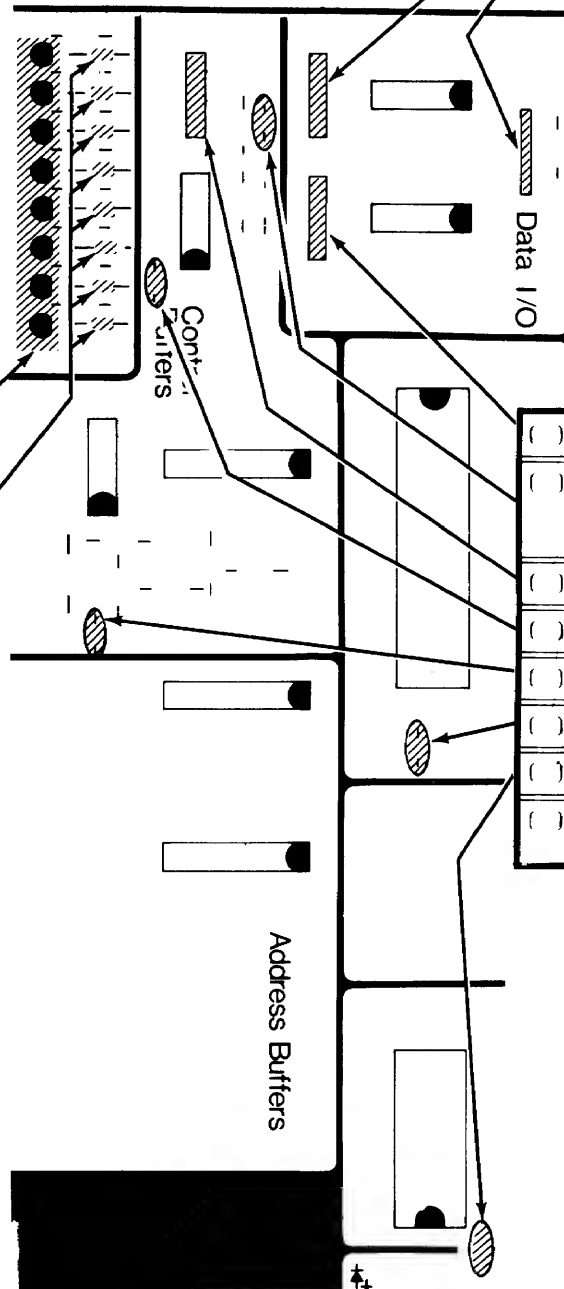
() LED2 through LED9: Install eight 1/4" red LED's in the shaded area. Solder the leads to the foil and cut off the excess lead lengths.

() R32 through R39: Install eight 180 Ω (brown-gray-brown) resistors. Solder the leads to the foil and cut off the excess lead lengths.

() Locate an 8-pin connector block. Then refer to Detail 1-8A below and check all four contacts on the bottom of the block. If you find any burrs or raised edges, press them down with a screwdriver blade or similar tool. This will prevent them from causing a short circuit on the circuit board. NOTE: Make sure you use this connector block in the next step.



Detail 1-8A



() 8-pin connector block.

() 4-pin connector block.

() 8-pin connector block.

() C19: .01 μ F ceramic. Solder the leads to the foil and cut off the excess lead lengths.

() 8-pin connector block.

() C21: .01 μ F ceramic.() C22: .01 μ F ceramic.() C18: .01 μ F ceramic.() C17: .01 μ F ceramic.

() Solder the leads to the foil and cut off the excess lead lengths.

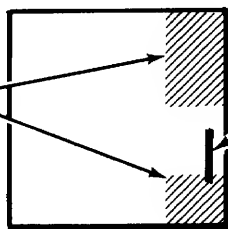
PICTORIAL 1-8



IDENTIFICATION DRAWING

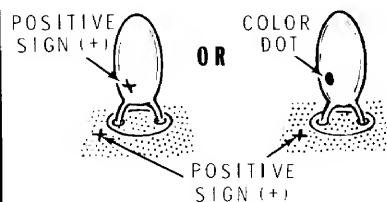
The steps performed in this Pictorial are in these areas of the circuit board.

PART
NUMBER



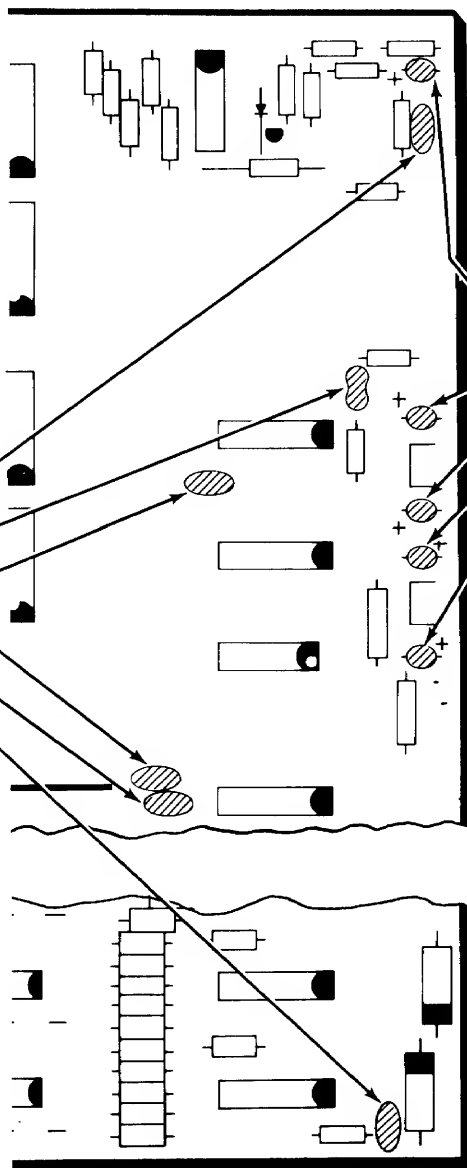
CONTINUE

NOTE: When you install a tantalum capacitor, install the lead marked with the positive (+) mark or color dot on the capacitor in the positive (+) marked hole on the board.



START

- () C14: .01 μ F ceramic.
- () C23: 100 pF mica.
- () C24: .01 μ F ceramic.
- () C15: .01 μ F ceramic.
- () C4: .01 μ F ceramic.
- () C16: .01 μ F ceramic.
- () Solder the leads to the foil and cut off the excess lead lengths.



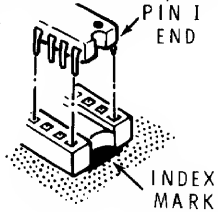
- () C13: .68 μ F tantalum.
- () C11: 10 μ F (10M) tantalum.
- () C8: 2.2 μ F tantalum.
- () C12: 10 μ F (10M) tantalum.
- () C9: 2.2 μ F tantalum.
- () Solder the leads to the foil and cut off the excess lead lengths.

PICTORIAL 1-9

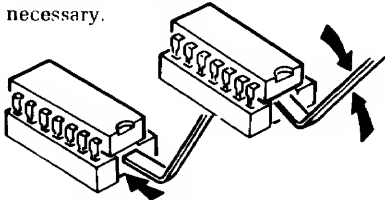
START ➔

In the following steps, install IC's in the designated sockets. Be careful to match the pin 1 end of each IC to the index mark on the circuit board. See Detail 1-10A.

Before you apply downward pressure to an IC, make sure each IC pin is centered in its proper socket hole. Handle IC's with care, as their pins bend very easily.

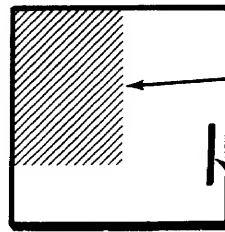


NOTE: An IC puller has been furnished to remove an IC from its socket if necessary.

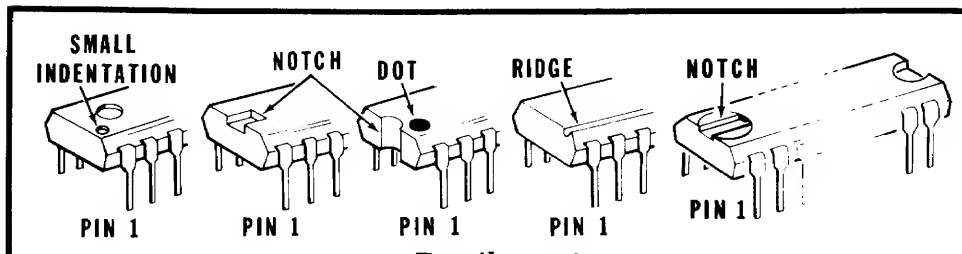
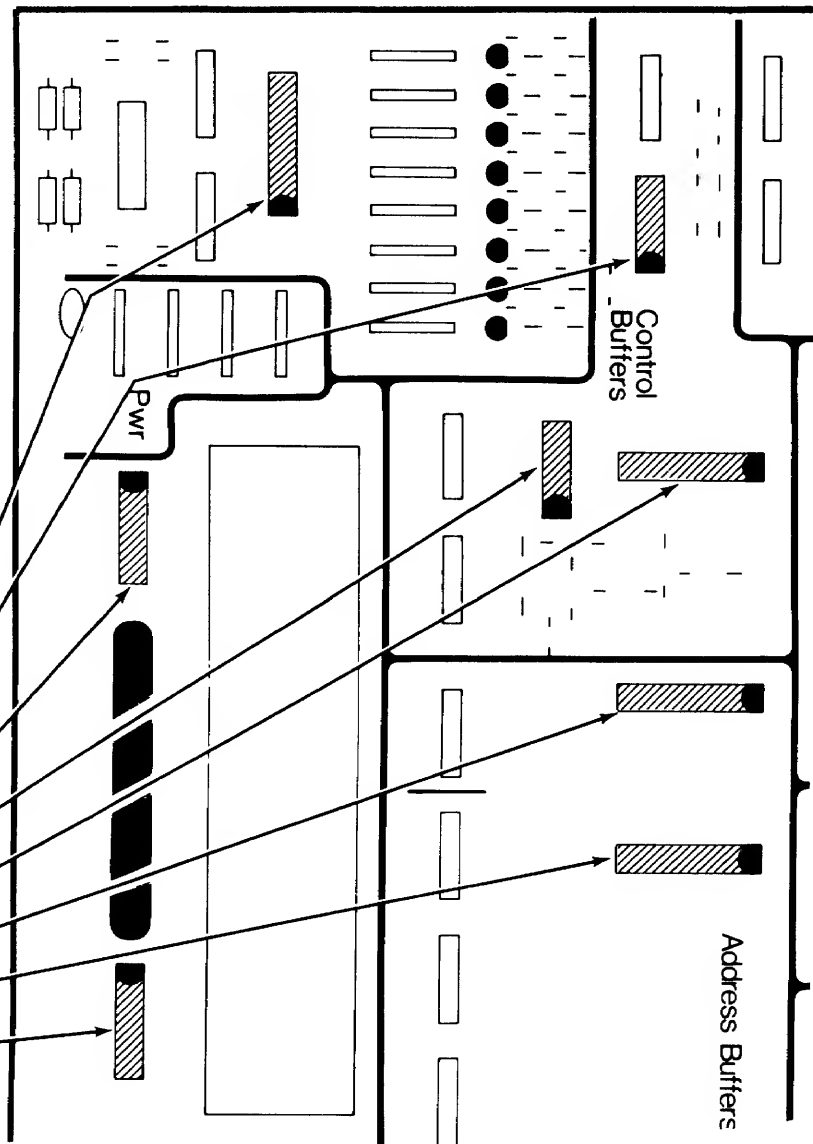


Push the shorter end of the puller in between the IC and the socket and rock the longer portion back and forth. Be very careful, as the IC pins are very easily bent.

- () IC1: 74LS241 (#443-824).
- () IC4: 74126N (#443-717).
- () IC2: 74LS42 (#443-807). Be sure to notice the index mark on the circuit board.
- () IC5: 74S00 (#443-26).
- () IC6: 74LS241 (#443-824).
- () IC7: 74LS241 (#443-824).
- () IC8: 74LS241 (#443-824).
- () IC3: 74LS42 (#443-807).



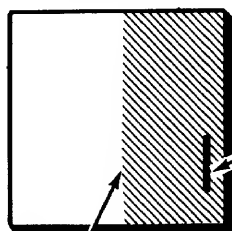
The steps performed in this Pictorial are in this area of the circuit board.



Detail 1-10A.

PICTORIAL 1-10

IDENTIFICATION DRAWING

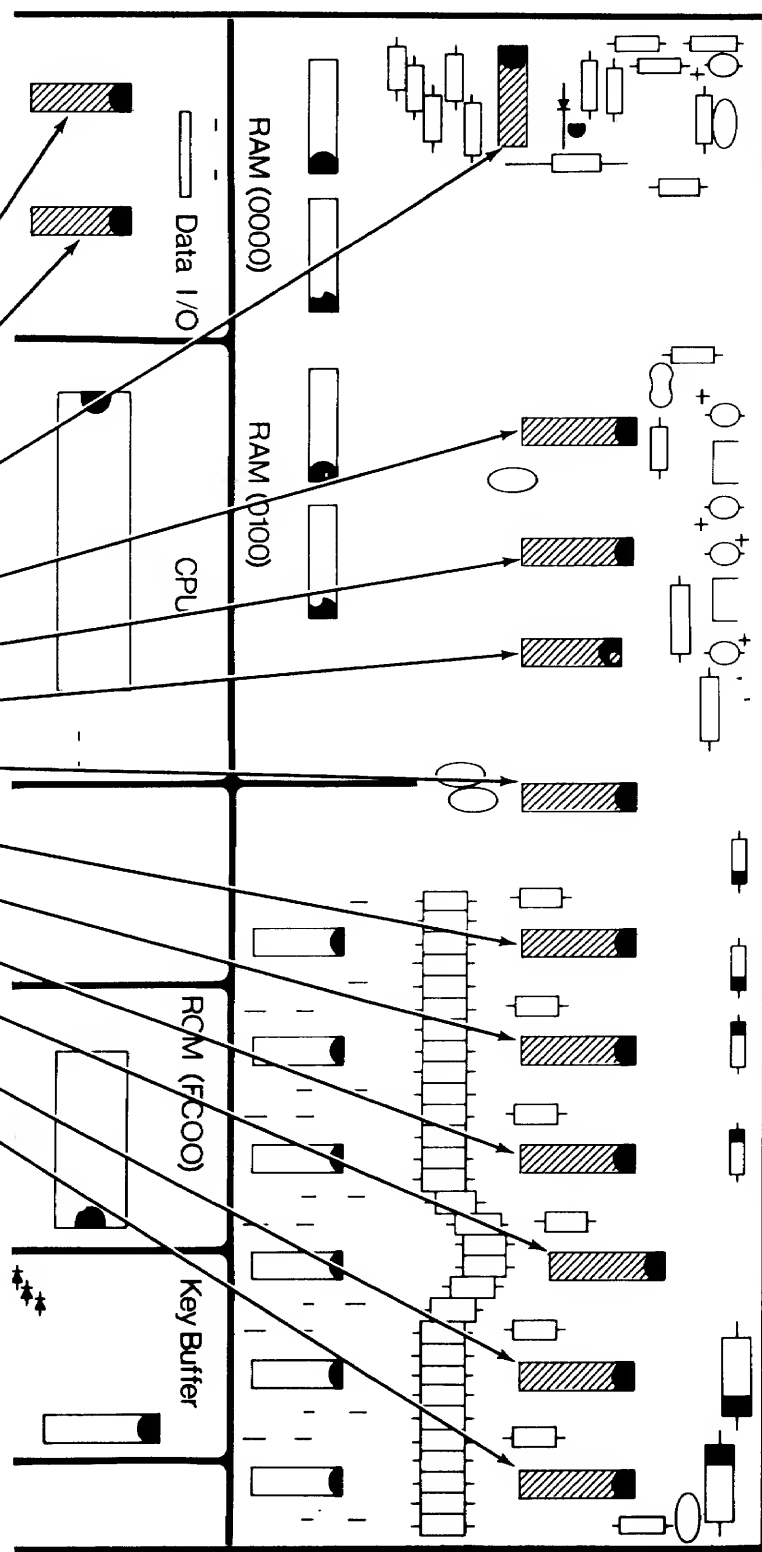

PART
NUMBER

The steps performed in this Pictorial are in this area of the circuit board.

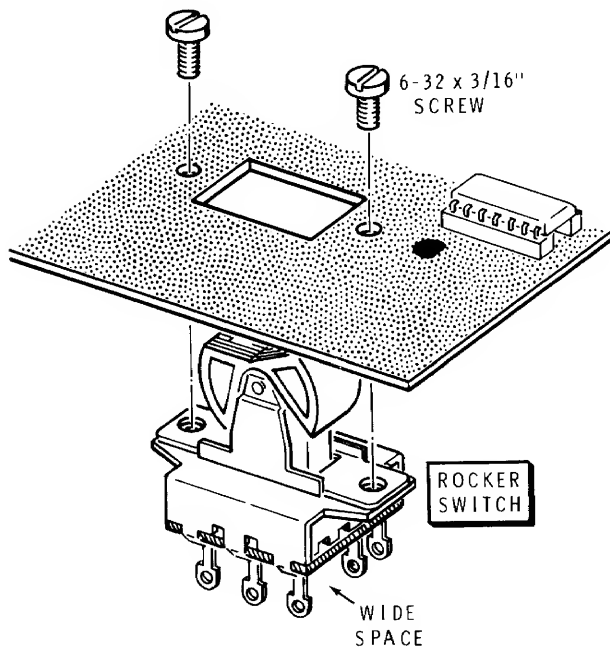
START

- () IC9: 74LS243 (#443-839).
- () IC10: 74LS243 (#443-839).
- () IC18: LM3302N or LM2901N (#442-616).
- () IC19: MC6875 (#443-840).
- () IC20: 74LS42 (#443-807).
- () IC21: 74S00 (#443-26).
- () IC22: 74LS42 (#443-807).
- () IC23: 74LS259 (#443-804).
- () IC24: 74LS259 (#443-804).
- () IC25: 74LS259 (#443-804).
- () IC26: 74LS259 (#443-804).
- () IC27: 74LS259 (#443-804).
- () IC28: 74LS259 (#443-804).

NOTE: The remaining integrated circuits will be installed later. IC's 16 and 17 are supplied with the educational course.

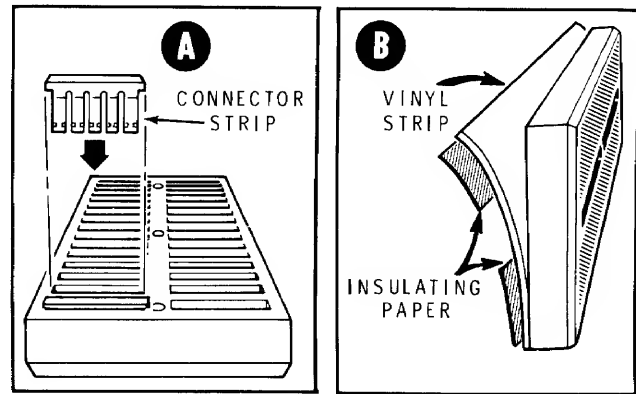


PICTORIAL 1-11

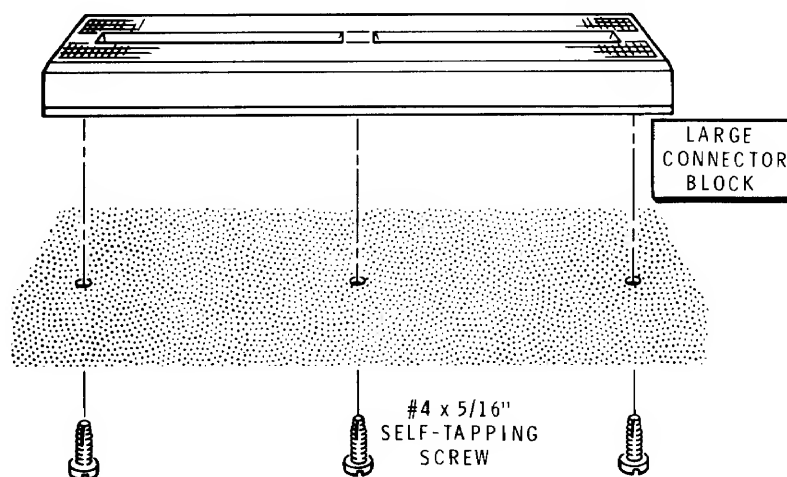
**Detail 1-12A**

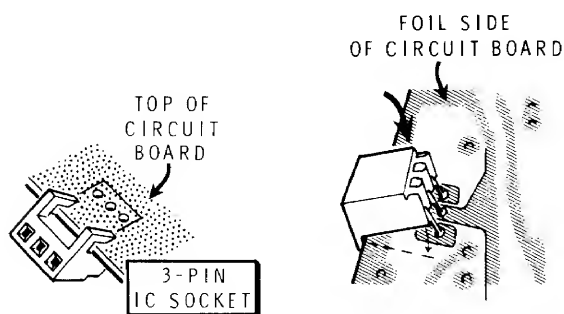
Refer to Pictorial 1-12 (Illustration Booklet, Page 3) for the following steps.

- () Reposition the main circuit board as shown.
- () SW1: Refer to Detail 1-12A and mount the rocker switch on the main circuit board at SW1 with two 6-32 \times 3/16" screws. Install the switch so the lugs are positioned as shown in the Detail.

**Detail 1-12B**

- () Refer to Part A of Detail 1-12B and install the connector strips (supplied with the large connector block) into the block in the manner shown. NOTE: You may have some connector strips left over.
- () Turn the connector block right side up, and with a screwdriver handle or similar tool, tap on the top of the block until all the connector strips are fully seated up into the block.
- () Refer to Part B of Detail 1-12B and remove the paper backing from the vinyl strip supplied with the connector block. Position the connector as shown, line up the long edge of the vinyl strip with the long edge of the connector block, and firmly press the strip onto the block.
- () Refer to Detail 1-12B and remove the backing paper from the insulating paper. Then apply the insulating paper along the indicated edges of the vinyl strip. Keep the paper even with the edges of the large connector block.

**Detail 1-12C**



Detail 1-12D

- () With the tip of a pencil, push through the three mounting hole locations in the vinyl strip.
- () Refer to Detail 1-12C and mount the large connector block on the main circuit board with three #4 × 5/16" self-tapping screws.
- () Refer to Detail 1-12D and mount a 3-pin IC socket at IC29 on the top edge of the circuit board as shown. Place the edge-mount retainers over the edge of the circuit board; then rotate the pins into their holes on the **foil side** of the board. Carefully solder the three pins to the foil.
- () In the same manner, install a 3-pin IC socket at IC30.
- () LED1: Refer to Detail 1-12E and mount the 3/8" red LED near the rocker switch as shown. Be sure to match the flat on the LED with the outline of the flat on the circuit board. NOTE: Before you cut off the excess leads, be sure the bottom edge of the LED is 1/4" above the board, and that it is not tilted. Solder the leads to the foil.

CIRCUIT BOARD CHECKOUT

Carefully inspect the foil side of the circuit board for the following conditions.

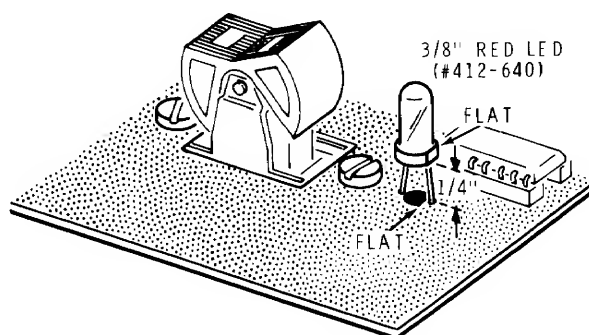
- () Unsoldered connections.
- () Poor solder connections.
- () Solder bridges between foil patterns. NOTE: If you are in doubt about a foil pattern, refer to the "Circuit Board X-Ray View" (Illustration Booklet, Page 18).
- () Protruding leads which could touch together.

Carefully inspect the component side of the circuit board for the following conditions.

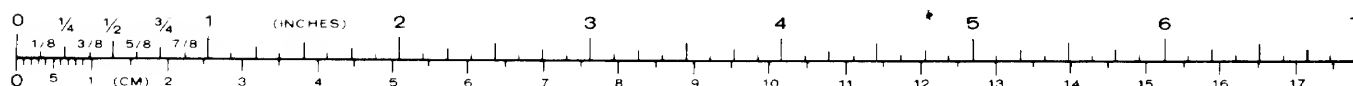
- () Integrated circuits for proper type and installation.
- () Tantalum capacitors for the correct position of the positive (+) mark or dot.
- () Diodes for the correct position of the banded ends.
- () LED's for the correct position of the flat sides.

NOTE: There are many unused connections on the foil side of the main circuit board, some of which will be used later. As you make further connections to the circuit board, be sure to inspect each one carefully to be sure the foils remain unbridged.

Set the main circuit board aside temporarily.



Detail 1-12E



KEYBOARD CIRCUIT BOARD

START ➔

Position the keyboard circuit board as shown. Then proceed with the following steps.

NOTE: To prepare a wire, as in the following step, cut it to the indicated length and remove 1/4" of insulation from each end. If the wire is stranded, tightly twist each wire end and apply a small amount of solder to hold the fine strands together.

() Prepare the following wires:

One 2-1/2" white stranded
Nine 1-1/2" white stranded
One 1-3/8" yellow

As you install a prepared wire in the following steps, solder it to the foil and cut off the excess wire length.

() 2-1/2" white wire at A.

() Nine 1-1/2" white wires at B through K.

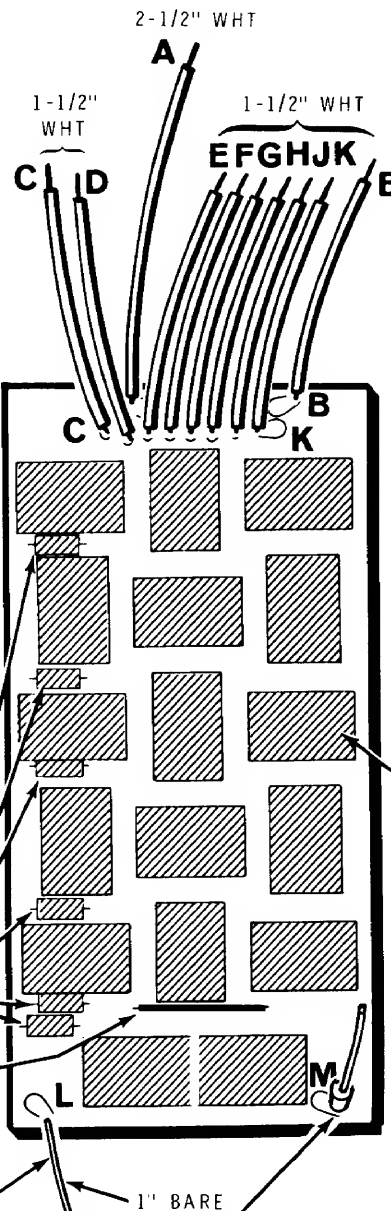
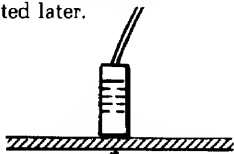
() R52 through R57: Install six 8200 Ω (gray-red-red) resistors. Solder the leads to the foil and cut off the excess lead lengths.

() 1-3/8" yellow wire jumper.

() Remove the insulation from 1" of brown wire. Then cut off this bare wire.

() 1" bare wire at L.

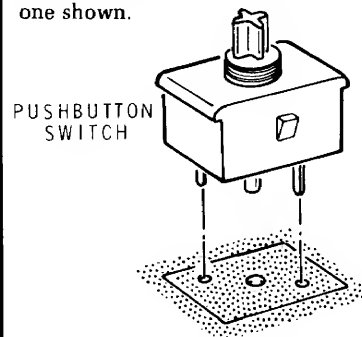
() R107: 180 Ω (brown-gray-brown). Mount it vertically down on the circuit board, solder the lead to the foil, and cut off the excess lead length. The free lead will be connected later.



PICTORIAL 2-1

CONTINUE ➔

As you install pushbutton switches in the following step, be sure each key is down against the top of the keyboard before you solder its two lugs. Your switches may look different than the one shown.



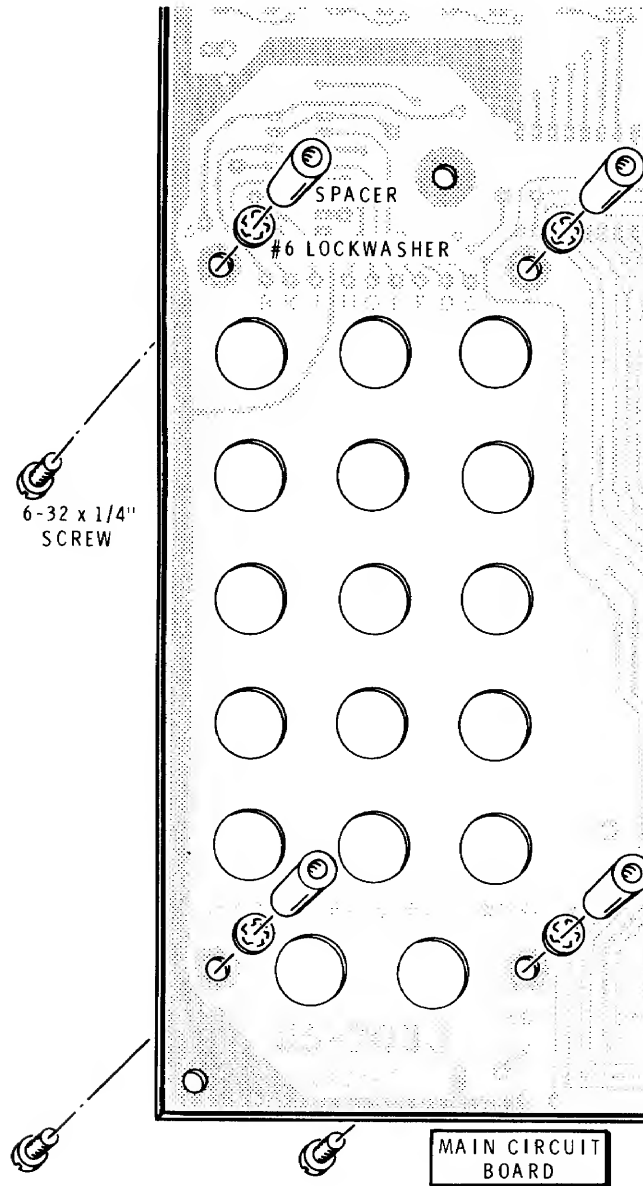
() 17 pushbutton switches.

CIRCUIT BOARD CHECKOUT

Carefully inspect the circuit board for the following conditions.

- () Unsoldered connections.
- () "Cold" solder connections.
- () Solder bridges between foil patterns.
- () Protruding leads which could touch together.

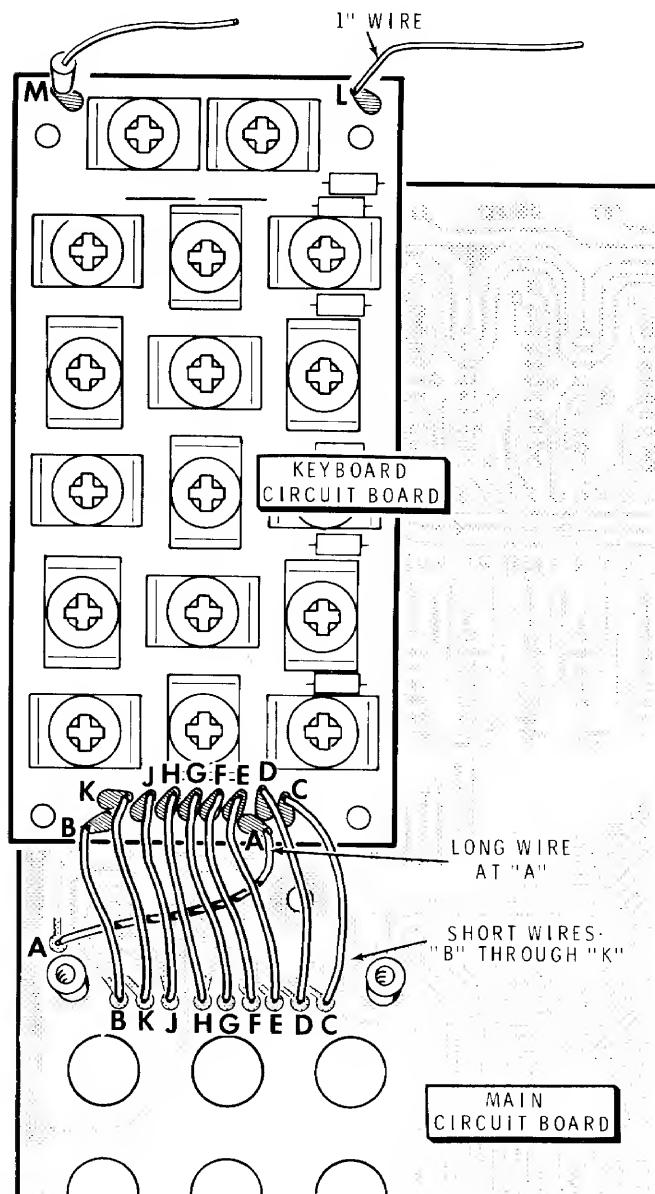




Detail 3-1A

Refer to Pictorial 3-1 (Illustration Booklet, Page 3) for the following steps.

- () Refer to Detail 3-1A, turn the main circuit board upside down, and loosely mount spacers onto the foil side at the four locations shown in the Pictorial. Use 6-32 \times 1/4" screws and #6 lockwashers.



Detail 3-1B

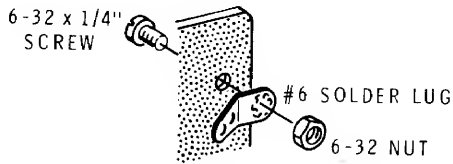
- () Position the keyboard circuit board, foil-side down, onto the main circuit board as shown in Detail 3-1B.

NOTE: As you install each wire, push it into its designated hole and leave approximately 1/16" of the bare wire above the foil so it will make a good solder connection. Solder each wire to the foil as it is installed and cut off the excess wire ends on the top of the circuit board.

Connect the wires coming from the keyboard circuit board to the main circuit board:

- () Wire B to B.
- () Wire K to K.
- () Wire J to J.
- () Wire H to H.
- () Wire G to G.
- () Wire F to F.
- () Wire E to E.
- () Wire A to A.
- () Wire D to D.
- () Wire C to C.
- () Flip the keyboard circuit board over, end-for-end, (keep the wires out of the way) and position the tops of the pushbutton switches into their corresponding holes in the main circuit board. (If your switches have springs and brass washers, you may have to force them through the holes).
- () Connect the wire coming from keyboard hole L to hole M on the main circuit board. **Do not** solder the connection.
- () Connect the resistor coming from keyboard hole M to hole L on the main circuit board. **Do not** solder the connection.
- () Loosely install four 6-32 \times 1/4" screws and #6 lockwashers at the keyboard corner holes. Turn the screws into the spacers as shown.
- () On the top of the main circuit board, tighten the four 6-32 screws to secure the spacers; then tighten the four keyboard mounting screws.
- () Solder the wire and resistor lead to the main circuit board at L and M and cut off the excess lengths.





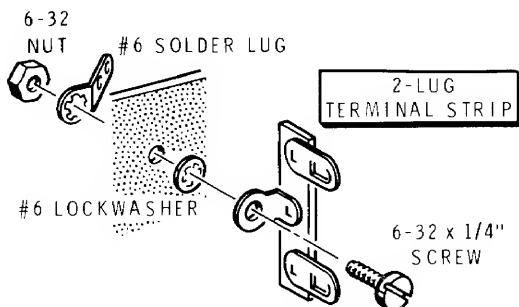
Detail 4-1A

SUPPORT BRACKET ASSEMBLY

Refer to Pictorial 4-1 (Illustration Booklet, Page 4) for the following steps.

- () Position the support bracket on your work area as shown.
- () Refer to Detail 4-1A and mount a solder lug at A with a 6-32 \times 1/4" screw and a 6-32 nut. Position the solder lug as shown in the Pictorial.
- () Press a rubber grommet into hole B.
- () Refer to Detail 4-1B and mount a 2-lug terminal strip at C. Use a 6-32 \times 1/4" screw, #6 lockwasher, #6 solder lug, and 6-32 nut. Position the terminal strip and solder lug as shown in the Pictorial.
- () Cut the lead at the **positive** (+) end of a 1200 μ F electrolytic capacitor (#25-241) to 1/2".

NOTE: In the following steps, (NS) means not to solder a connection because other wires or leads will be connected later. "S-" with a number, such as (S-2),



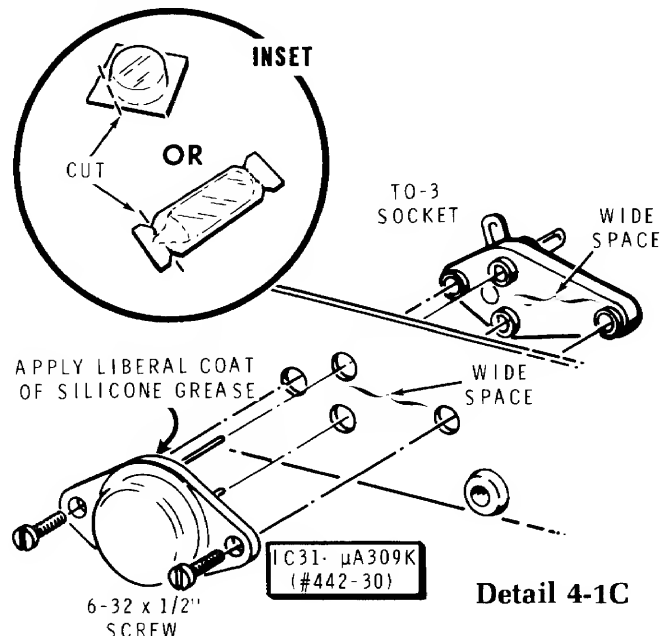
Detail 4-1B

means to solder the connection. The number following the "S" tells how many wires are at the connection.

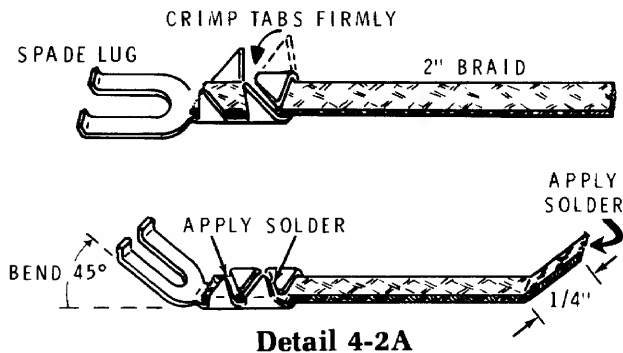
- () C6: Connect the **positive** (+) lead of a 1200 μ F capacitor to terminal strip C lug 1 (NS) and the negative lead to solder lug A (NS). Position the capacitor as shown in the Pictorial.
- () Cut the lead at the negative (unmarked) end of another 1200 μ F electrolytic capacitor to 1/2".
- () C7: Connect the **negative** (unmarked) lead of the other 1200 μ F capacitor to terminal strip C lug 2 (NS) and the positive (+) lead to solder lug A (S-2).

Refer to Detail 4-1C for the next two steps.

- () Refer to the inset drawing on Detail 4-1C and open the container of silicone grease. Apply a liberal coating of the grease to the bottom of the μ A309K integrated circuit (#442-30).
- () IC31: Carefully observe the wide spacing on the IC holes in the support bracket at IC31 and place the TO-3 socket on the underside of the bracket as shown in the Detail. Be sure the shoulders of the socket are centered in the two end holes. Then push the pins of the μ A309K IC into the socket, through the support bracket. Making sure the socket shoulders are still centered in their holes, secure the IC with two 6-32 \times 1/2" screws.

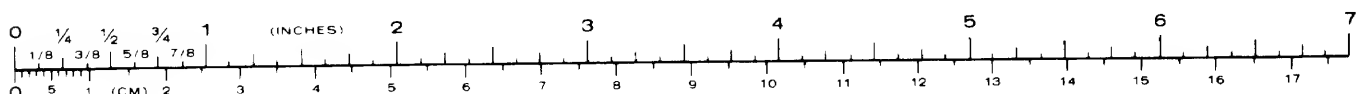
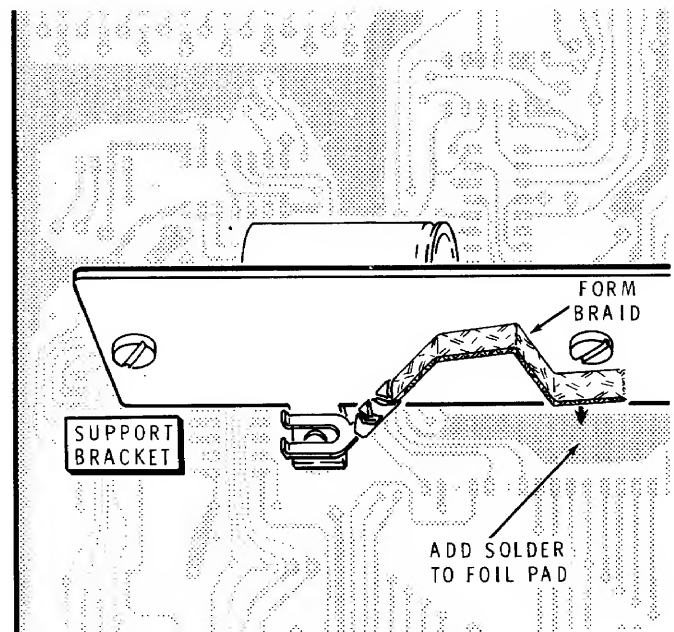


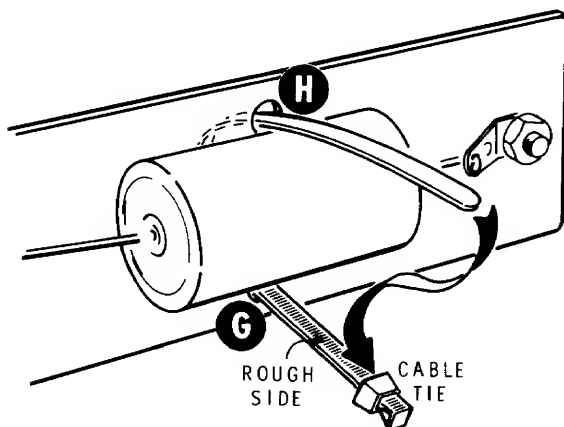
Detail 4-1C



Refer to Pictorial 4-2 (Illustration Booklet, Page 4) for the following steps.

- () Reposition the support bracket as shown.
- () As you did in Detail 4-1A, mount a #6 solder lug at D with a 6-32 \times 1/4" screw and a 6-32 nut. Position the solder lug as shown in the Pictorial.
- () Cut a 1-3/4" piece of sleeving.
- () C1: Cut the negative (unmarked) lead of the 6000 μ F electrolytic capacitor to 1". Connect the negative lead of the capacitor to solder lug D (S-1). Place the 1-3/4" sleeve on the positive (+) lead and connect the lead to socket IC31 lug 1 (NS).
- () Prepare a 3" yellow wire.
- () Connect the 3" yellow wire from solder lug C (S-1) to IC31 lug 3 (NS).
- () C2: Cut one lead of a .22 μ F Mylar capacitor to 5/8". Connect this shortened lead to socket IC31 lug 3 (NS) and the longer lead to lug 1 (NS).
- () C3: Cut one lead of the other .22 μ F Mylar capacitor to 5/8". Connect this lead to socket IC31 lug 3 (S-3) and the longer lead to lug 2 (NS).
- () Prepare a 7-1/4" orange wire.
- () Connect one end of the 7-1/4" orange wire to socket IC31 lug 2 (S-2).
- () Prepare a 9" white-orange wire.
- () Connect one end of the 9" white-orange wire to socket IC31 lug 1 (S-3).
- () Cut a 2" piece of flat braid.
- () Refer to Detail 4-2A and crimp and solder a spade lug onto one end of the 2" braid. Apply a liberal amount of solder to 1/4" of the free braid end.
- () Loosely mount the support bracket to the foil side of the main circuit board at E with a 6-32 \times 1/4" screw and a 6-32 nut. Secure the support bracket and the spade lug with the braid at F with a 6-32 \times 1/4" screw and 6-32 nut. Be sure to position the free end of the braid as shown in the Pictorial. Tighten the support bracket mounting hardware.
- () Refer to the Pictorial and form the center of the braid and the spade lug as shown to be sure the braid will not come in contact with any of the other circuit board foils.
- () Refer to Detail 4-2B and add a liberal amount of solder to the indicated foil pad on the main circuit board. Be very careful not to form a bridge to other foils. Press the free end of the braid onto the top of this foil pad and heat it with the soldering iron until the solder melts into the braid. Hold the braid in place with pliers until it has cooled.



**Detail 4-2C**

- () Refer to Detail 4-2C and pass the tip of the cable tie through hole G in the support bracket making sure the rough side is facing upward. Then pass the tie across the rear of the bracket and back through hole H making sure the rough side is down. Pass the cable tie around capacitor C1 and push the tip of the tie through the other end retainer as shown. Pull the tie until it is tightly secure around the capacitor; then cut off the excess tie end.

Refer to Pictorial 4-3 (Illustration Booklet, Page 5) for the following steps.

- () Reposition the circuit board as shown.
- () Prepare the following wires:

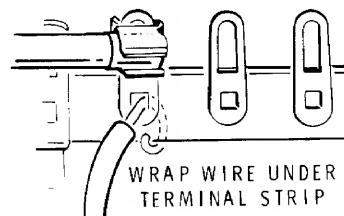
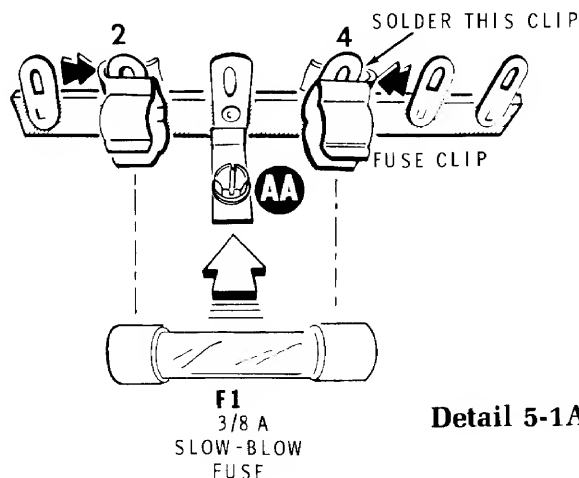
1-1/4" red	6-1/2" yellow
2-1/2" white-brown	6-1/4" white-yellow
2-1/2" brown	13" brown
1-3/4" orange	13" white-brown

NOTE: As you install wires in the following steps, form each of them as shown in the Pictorial. After a wire has been soldered to the foil or to the switch lug, cut off any excess wire lengths.

- () Connect one end of a 1-1/4" red wire to switch SW1 lug 1 (S-1). Slide a ferrite bead (#475-12) onto the free end of this wire; then connect the free end to the main circuit board hole R (S-1).

- () Connect a 2-1/2" white-brown wire from hole S (S-1) to switch SW1 lug 7 (S-1).
- () Connect a 2-1/2" brown wire from hole T (S-1) to switch SW1 lug 4 (S-1).
- () Connect a 1-3/4" orange wire from hole U (S-1) to switch SW1 lug 2 (NS). Be sure this wire does not cover the large nearby hole.
- () Form the orange wire coming from socket IC31 lug 2 downward and across the circuit board as shown. Connect the free end of the wire to SW1 lug 2 (S-2).
- () Route the free end of the white-orange wire coming from socket IC31 lug 1 downward to the board, and along the board as shown. Connect the free end of the wire to circuit board hole X (S-1).
- () Connect one end of a 6-1/2" yellow wire to circuit board hole W (S-1). Route the wire rearward, through support bracket grommet B. Connect the wire end to terminal strip C lug 1 (S-2).
- () Connect one end of a 6-1/4" white-yellow wire to circuit board hole V (S-1). Route the wire rearward, through support bracket grommet B. Connect the wire end to terminal strip C lug 2 (S-2).
- () Connect one end of a 13" brown wire to circuit board hole P (S-1). Route the wire forward, through support bracket grommet B. Connect the free end of the wire to switch SW1 lug 5 (S-1).
- () Connect one end of a 13" white-brown wire to circuit board hole N (S-1). Route the wire forward and through grommet B. Connect the free end of the wire to switch SW1 lug 8 (S-1).

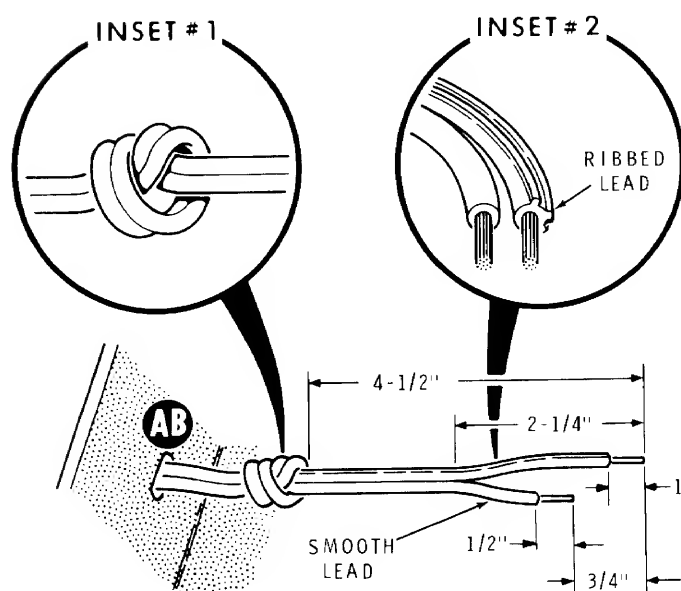
Set the main circuit board assembly aside temporarily.



CABINET ASSEMBLY AND WIRING.

Refer to Pictorial 5-1 (Illustration Booklet, Page 5) for the following steps.

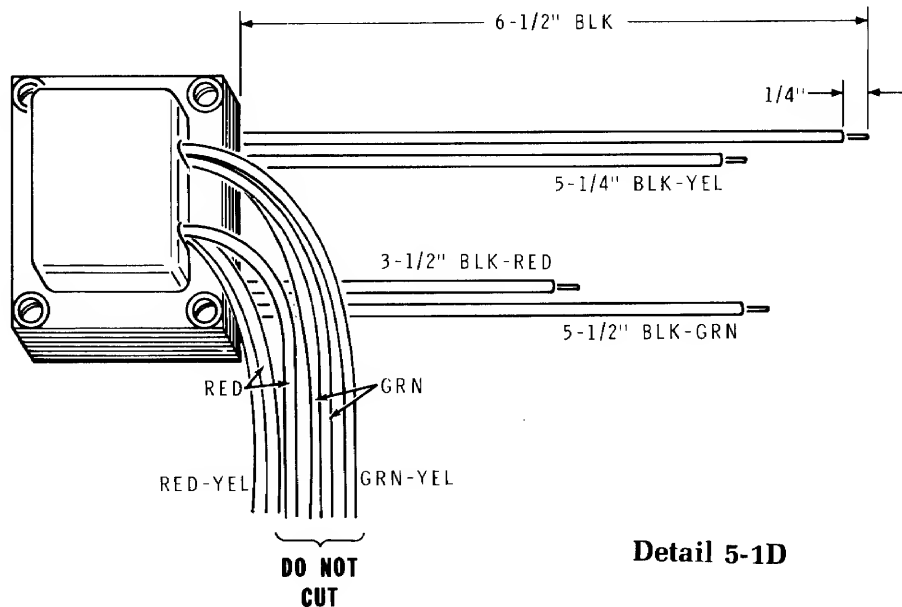
- () Temporarily mount a 6-lug terminal strip on cabinet post AA with a #6 × 3/8" hex head screw as shown.
- () F1: Refer to Detail 5-1A and install two fuse clips and the 3/8-ampere fuse on terminal strip AA lugs 2 and 4. Solder the fuse clip onto lug 4 only. NOTE: Do not use excessive heat to avoid damage to the fuse.
- () Refer to inset drawing #1 on Detail 5-1B and insert the end of the line cord through hole AB from the outside of the cabinet bottom. Tie a knot in the line cord 4-1/2" from the end as shown.
- () Refer to inset drawing #2 on Detail 5-1B and identify the smooth lead and the ribbed lead of the line cord. Then prepare the end of the line cord as shown in the Detail.
- () Tightly twist the bare wire ends and apply a small amount of solder to hold the fine strands together.



NOTE: As you connect the line cord leads in the following steps, be sure to make a mechanically secure connection. Wrap the lead ends securely under the terminal strip as shown in Detail 5-1C.

- () Smooth lead to the eyelet of lug 4 (S-1).
- () Ribbed lead to the eyelet of lug 6 (S-1).
- () Refer to Detail 5-1D and prepare the transformer leads as shown. Measure the leads from the edge of the transformer. If necessary, twist the lead ends tightly and apply a small amount of solder.
- () T1: Refer to Pictorial 5-1 and install the power transformer with the red and green leads up as shown. Use #6 × 1-1/8" self-tapping screws.



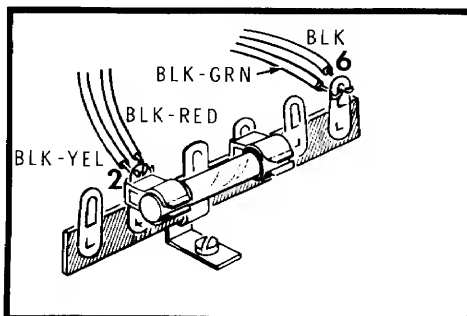


Detail 5-1D

ALTERNATE LINE VOLTAGE WIRING

Two sets of line voltage wiring instructions are given below, one for 120 VAC and the other for 240 VAC. In the United States, 120 VAC is most common. USE ONLY THE INSTRUCTIONS THAT AGREE WITH THE LINE VOLTAGE IN YOUR AREA.

FOR 120 VAC

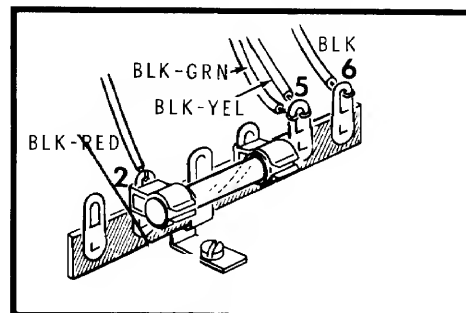


Detail 5-1E

Refer to Detail 5-1E for the following steps. In these steps, make connections to terminal strip AA. Wrap the lead ends tightly at the connections. Connect four of the power transformer leads as follows:

- () Black-red and black-yellow leads to lug 2 (S-2). NOTE: Also solder the fuse clip to lug 2.
- () Black-green and black leads to lug 6 (S-2).

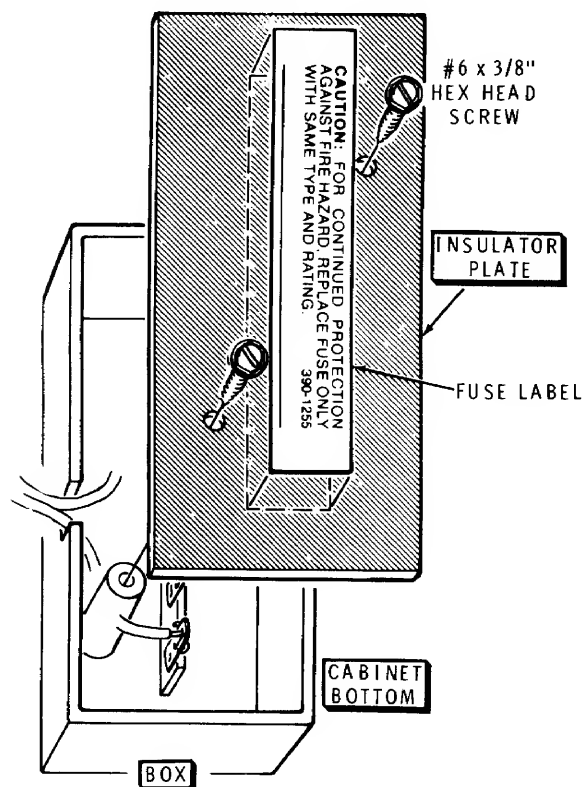
FOR 240 VAC



Detail 5-1F

Refer to Detail 5-1F for the following steps. In these steps, make connections to terminal strip AA. Wrap the lead ends tightly at the connection. Connect four of the power transformer leads as follows:

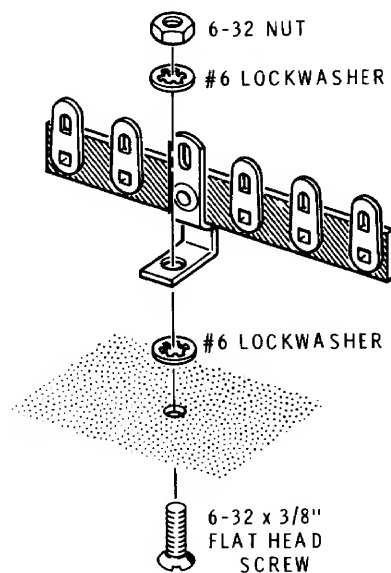
- () Black-red lead to lug 2 (S-1). NOTE: Also solder the fuse clip to lug 2.
- () Black-yellow and black-green leads to lug 5 (S-2).
- () Black lead to lug 6 (S-1).



PICTORIAL 5-2

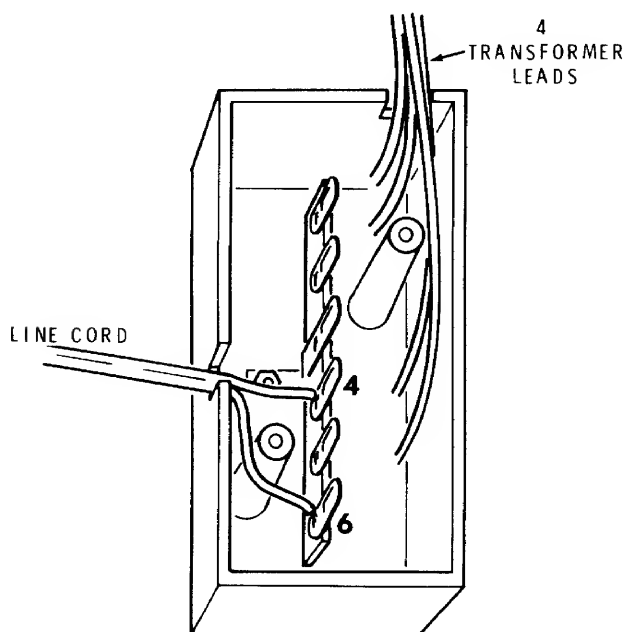
Refer to Pictorial 5-2 for the following steps.

- () Remove the fuse from the fuse clips. Then remove the screw you used to secure the terminal strip to the cabinet post.



Detail 5-2A

- () Refer to Detail 5-2A and mount the terminal strip in the box formed in the cabinet bottom as shown. Use a 6-32 \times 3/8" flat head screw, two #6 lockwashers, and a 6-32 nut. Position the terminal strip as shown in Detail 5-2B.
- () Reinstall fuse F1 in its fuse clips.

**Detail 5-2B**

- () Refer to Detail 5-2B and route the leads and wires as shown.
- () Mount the insulator plate to the terminal strip box with two #6 \times 3/8" hex head screws. Do not pinch any leads between the plate and the box.
- () Remove the paper backing from the fuse label and press the label in place onto the insulator plate. Then write the fuse information on the label: "3/8-Amp, 3AG, slow-blow."

Again, refer to Pictorial 5-1 (Illustration Booklet, Page 5) for the following steps.

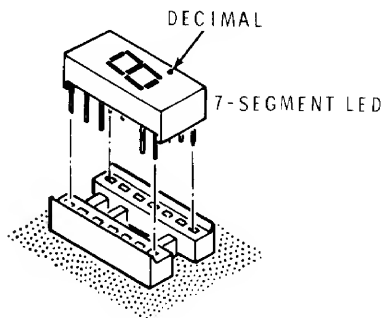
NOTE: As you connect each of the remaining power transformer wires to the main circuit board in the following steps, solder the lead to the foil and cut off the excess lead lengths.

- () Position the main circuit board, component side up, near the power transformer as shown in Pictorial 5-1.
- () Connect the red-yellow transformer lead to the circuit board hole labeled "RED/YEL."
- () Connect the green-yellow lead to the hole labeled "GRN/YEL."
- () Connect either red lead to one hole labeled "RED."
- () Connect the other red lead to the remaining "RED" hole.
- () Connect one green lead to one of the holes labeled "GRN."
- () Connect the other green lead to the remaining "GRN" hole.

NOTE: The remaining yellow wire is for any experiments you may want to do.

This completes the "Step-by-Step Assembly." Proceed to "Initial Tests."

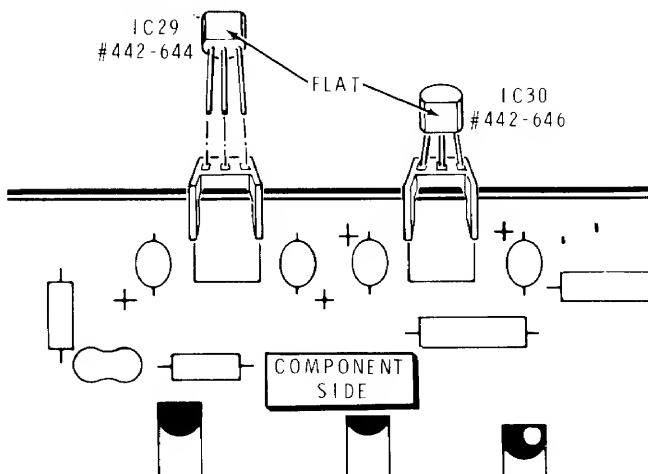
INITIAL TESTS



Detail 6-1A

Refer to Pictorial 6-1 (Illustration Booklet, Page 6) for the following steps.

- () Position the main circuit board part way out of the cabinet bottom as shown.
- () Refer to Detail 6-1A and carefully install a 7-segment LED at "H" in the manner shown. **NOTE:** Be sure a decimal point is at the **bottom right** as shown in the Pictorial. (Do not shorten the leads. They act as heat sinks.)
- () In the same manner, install the remaining five 7-segment LED's at "I," "N," "Z," "V," and "C."
- () IC29: Refer to Detail 6-1B and carefully install the LM78L12 IC (#442-644) into the socket at IC29 in the manner shown.



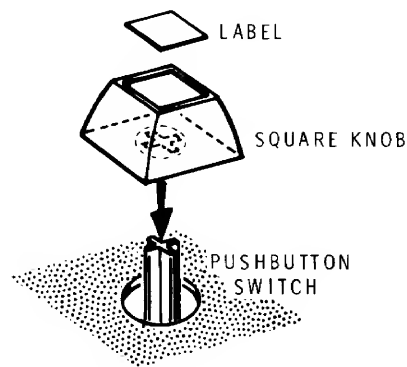
Detail 6-1B

VOLTAGE TESTS

NOTE: If at any time during the following tests you fail to obtain the desired results, and if power is applied to the unit, immediately unplug the line cord from the outlet and refer to the "In Case of Difficulty" section on Page 76.

You will need a volt-ohmmeter to perform the following tests. If such a meter is not available, proceed to "Tests Continued."

- () Connect one ohmmeter lead to one prong of the line cord plug, and the other lead to the remaining prong. The ohmmeter reading should be near or at zero.
- () Push down on the left side of the POWER switch (SW1) to be sure it is Off.
- () Plug the line cord into an AC outlet. The red LED next to the power switch should come on immediately and will remain on, regardless of the power switch setting.
- () Prepare two 1-1/2" wires. These may be of any color.
- () Locate the 4-pin connector blocks near the lower left corner of the circuit board labeled "+5" and "GND." Push one end of a short wire into each of these blocks.
- () Set your voltmeter to read +5 volts. Connect the positive lead to the wire at "+5" and the negative lead to "GND."
- () Push down on the right side of the POWER switch (SW1).
- () You should read 4.5 to 5.5 volts on the voltmeter.



Detail 6-2A

- () Set the voltmeter to read +12 volts. Move the positive meter lead and the test wire from "+5" to "+12." You should read 10.8 to 13.2 volts on the meter.
- () Push the POWER switch to Off and remove the line cord plug from the AC outlet.

- () Remove the voltmeter leads from the test wires; then move the test wire at "+12" to "-12."

Refer to Pictorial 6-2 (Illustration Booklet, Page 7) for the following steps.

- () Connect the positive test lead to "GND" and the negative lead to "-12." You should read 10.8 to 13.2 volts on the meter.

- () Refer to Detail 6-2A and place a square knob onto one of the pushbutton switches at the lower right portion of the circuit board. Push firmly on the knob to seat it onto the switch.

This concludes the portion of the tests that require the use of the volt-ohmmeter. Set the meter and wires aside.

- () In the same manner, install the remaining 16 square knobs on the pushbutton switches.

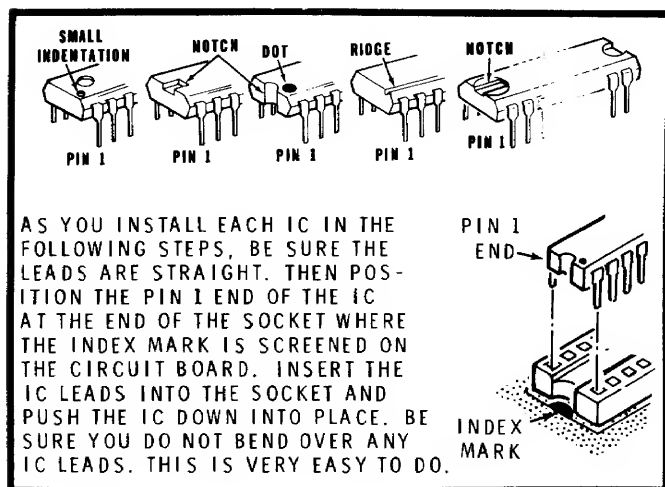
TESTS CONTINUED

- () If not already done, plug in the line cord and push down on the right side of the POWER switch (SW1). The red LED next to the POWER switch should turn on. (This LED will be on no matter which position the switch is in.)

- () Locate the keyboard label set. Then, one at a time, remove each of the numbered or lettered labels from the paper backing and press the label onto its correct pushbutton knob as shown in the Pictorial.

- () At the right edge of the circuit board, locate the "SEGMENT TEST" pins. Short these two pins together and note that all seven segments on the 7-segment LED's are lit, as well as the decimal point at the lower right of each LED. (Some LED's may already be lit.)

- () Locate the red label set. One at a time, remove the red labels from the paper backing, then position the label squarely over the 7-segment LED and press it in place. (You should have two labels left over.)

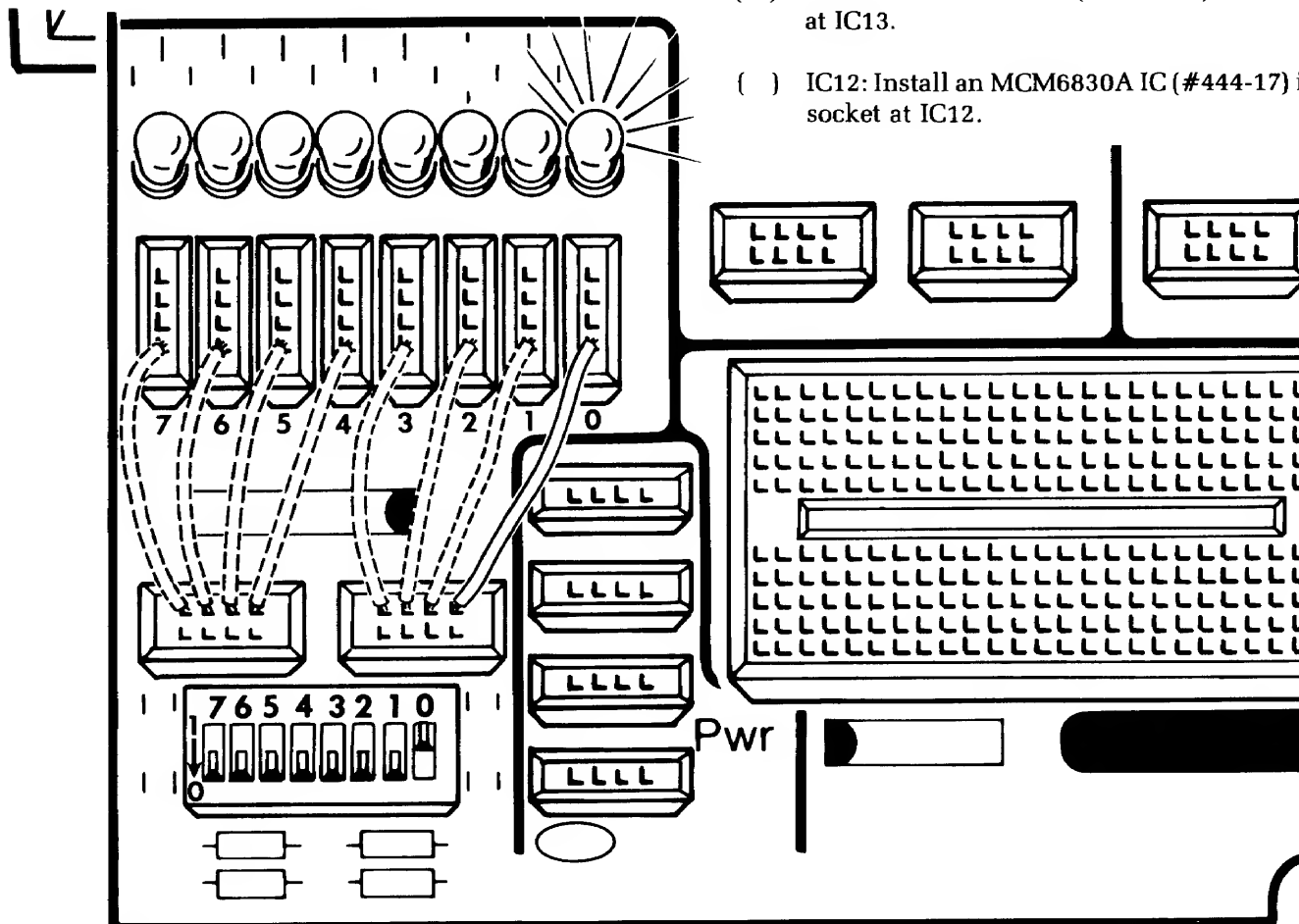


Detail 6-2B

NOTE: In the following steps, when you install an integrated circuit, refer to Detail 6-2B, remove the IC from its packing material (if necessary), and install the IC as shown.

Some of the IC's are packed in conductive foam. (Save this foam in case you ever remove these IC's.) These IC's are rugged, reliable components. However, normal static electricity discharged from your body through an IC pin to an object can damage the IC. Install these IC's without interruption as follows:

1. Remove the IC from its package with both hands.
 2. Hold the IC with one hand and straighten any bent pins with the other hand.
 3. Refer to Detail 6-2B. Position the pin 1 end of the IC over the index mark on the circuit board.
 4. Be sure each IC pin is properly started into the socket. Then push the IC down.
- () IC14: Install a 2112-2 IC (#443-721) in the socket at IC14.
- () IC15: Install a 2112-IC (#443-721) in the socket at IC15.
- () IC13: Install a 40097 IC (#443-720) in the socket at IC13.
- () IC12: Install an MCM6830A IC (#444-17) in the socket at IC12.



Detail 6-2C



- () IC11: Install an MC6800P IC (#443-827) in the socket at IC11.
- () Prepare a 4" yellow wire.
- () Plug in the line cord and turn the Trainer on.

Refer to Detail 6-2C for the following three steps.

NOTE: In the following steps, you will check out the Binary Data LED's at the lower left side of the circuit board. Each of these LED's is numbered (from right to left), directly beneath their corresponding 4-pin connector blocks, from "0" to "7." In addition, these connectors and LED's have corresponding switches on the slide switch assembly and pairs of connector pins in the two 8-pin connector blocks located immediately above the slide switch assembly.

- () Connect the 4" jumper wire from 4-pin connector block No. "0" to 8-pin block pair "0" (as shown on the Detail). Operate slide switch "0" and observe that the furthest right (zero) LED turns on and goes out.
- () Move the jumper wire to the "1" connector blocks, second from the right. Operate the slide switch and observe that the "1" LED turns on and goes out.
- () Progressively, and in the same manner, move the jumper wire to the "2," the "3," the "4," the "5," the "6," and the "7" connector blocks. Each time, operate the corresponding slide switch and observe that the correct LED is lit. Then remove the wire.

OPERATIONAL TESTS

This section of the Manual will check the basic Microprocessor functions to make sure they are working properly. The entries that will be made on the keyboard are not necessarily related to the actual use of the unit. Actual use of each function is explained in detail in the "Operation" section, starting on Page 45.

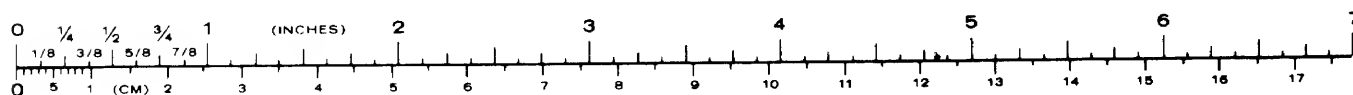
Refer to Pictorial 6-3 (Illustration Booklet, Page 7) to identify the function of each keyboard key.

NOTE: If you encounter any trouble in the following steps, turn the power off and remove the line cord plug from the AC outlet. Then refer to the "In Case of Difficulty" section on Page 91.

Each number step in the following charts shows which number or letter key to push, and what the resultant readout will be. Always push the keys in the sequence shown.























The following abbreviations are used on the Microprocessor keyboard:

ACCA	Accumulator "A"
ACCB	Accumulator "B"
PC	Program Counter
INDEX	Index Register
CC	Condition Codes Register
SP	Stack Pointer
RTI	Return From Interrupt
SS	Single Step
BR	Break Point
AUTO	Automatic Load
BACK	Back
CHAN	Change
DO	Do
EXAM	Examine
FWD	Forward



NOTES:

1. In the following charts, the symbol "*" is used to denote a blank readout indication. The symbol "X" indicates a random figure.
2. When you make two-digit entries, the indicated Readout display will be shown after the second digit key has been released.
3. If you make an incorrect entry, return to step 1.

STEP	FIRST PRESS:	THEN PRESS:	READOUT **
1			CPU * UP.
2			--- Ad.
3		   	0 123 XX
4			0 123 _ _
5		 	0 12345
6			--- Ad.
7		   	6 789 XX
8			6 789 _ _
9		 	6 789 AB
10			--- Ad.
11		   	C d E F XX

** Shows readout as presented on LED's after the key is pressed.

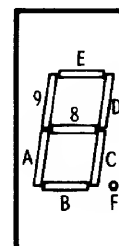


You have now determined that the Microprocessor keys are operating properly. Continue the operational test as you enter the following simple program.

STEP	FIRST PRESS:	THEN PRESS:	READOUT
1	RESET		<i>CPU* UP.</i>
2	AUTO A		<i>---- Ad.</i>
3		0 0 0 0 (Memory address)	<i>0000 _ _</i>
4		SS 8 SP 6 (Load Accumulator A w/following two characters)	<i>0001 _ _</i>
5		0 ACCA 1 (Data for first step)	<i>0002 _ _</i>
6		BACK B RTI 7 (Store Accumulator A at extended address of following four characters)	<i>0003 _ _</i>
7		CHAN C ACCA 1	<i>0004 _ _</i>
8		SP 6 FWD F	<i>0005 _ _</i>
9		RTI 7 EXAM E (Jump to extended address of following four bytes of information)	<i>0006 _ _</i>
10		0 0	<i>0007 _ _</i>
11		0 0	<i>0008 _ _</i>
12	RESET	(To terminate AUTO addressing sequence)	<i>CPU* UP.</i>
13	DO D		<i>---- do.</i>
14		0 0 0 0	<i>. * * * *</i>
15	RESET		<i>CPU* UP.</i>

The preceding program entered information into memory storage that told the Microprocessor that you wished to turn on the decimal point of the "H" LED. The program was proved in steps 13 and 14 above.

















Note in step 8 the characters "6F." This is the information that told the Microprocessor that you wished to address the "H" LED, and in particular, the decimal point; the "6" addressed the LED and the "F" addressed the decimal point. Refer to Detail 6-3A and note that each segment of an LED may be similarly addressed. Thus, to turn on each segment of the "H" (or "6") LED in turn, the terminal character must be changed to agree with the segment address.



7-SEGMENT
LED

Detail 6-3A

In the following chart, the top bar of the "H" LED will be addressed and examined.

STEP	FIRST PRESS:	THEN PRESS:	READOUT
1			CPU*UP.
2			---Ad.
3		   	00046F
4			0004--
5		  : "H" LED, TOP BAR.	00046E
6			CPU*UP.
7			---d.
8		   	- * * * *
9			CPU*UP.



As you observed in step 8, only the top-bar segment of the "H" LED lit up. You may further address and call up the remaining segments, in turn, of the same LED as follows: Repeat all nine steps in the preceding chart, with **one** exception. After you push the "CHAN/C" key in step 4, enter "6D;" then proceed with the remaining steps 6 through 8. The next time, at step 5, enter "6C", and proceed. In the same manner, at step 5 of each repetition, enter "6B, 6A, 69 and 68." Refer to Detail 6-3A to determine which LED segment should be lit.

To address the individual segments of the "I" LED, use the preceding chart and perform steps 1 through 4 as before. At step 5, enter "5F," and proceed with steps 6 through 8. Note that the decimal on the "I" LED will light. Then, at step 5, one at a time, enter "5E, 5D, 5C, 5B, 5A, 59, and 58." All segments of the remaining four LED's may be called up in a like manner using, for example: "4F, 3F, 2F, 1F" at step 5 to light the respective decimal segments.

This completes the "Operational Tests" of your Microprocessor Trainer.

NOTE:

Provision has been made for you to install a 40-pin connector for system expansion. Brief instructions

and a list of manufacturers are given below. If you do not wish to install a connector at this time, proceed directly to "Final Assembly" on Page 42.

Purchase and install a 40-pin connector* on the circuit board between IC2 and IC3. Then connect eight wires from the eight circuit board holes that connect the connector data pins to the eight data holes near IC9 and IC10. (See the "Schematic" and "Circuit Board X-Ray View.") These wires connect the eight data lines (D0-D7). Be sure you connect these wires properly so that data D0 goes to data line D0, etc.

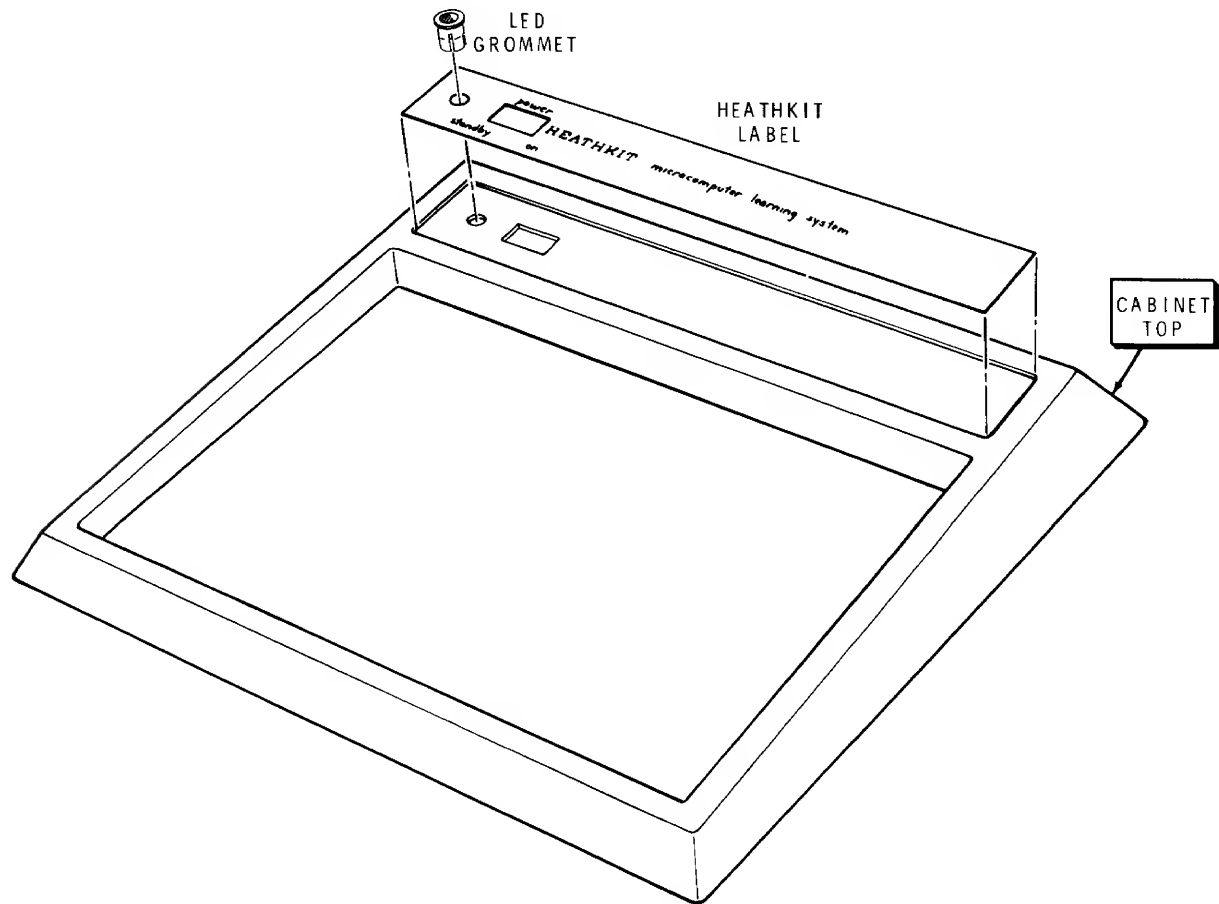
*The connectors must have .025" square pins on .100" centers. The following manufacturers supply such connectors. Some are single strips of connectors that must be cut to length.

AP Products
929834-01 (2 strips required)
929836-01

Molex
22-04-2201 (2 strips required)

AMP
2-87215-0
2-87543-0

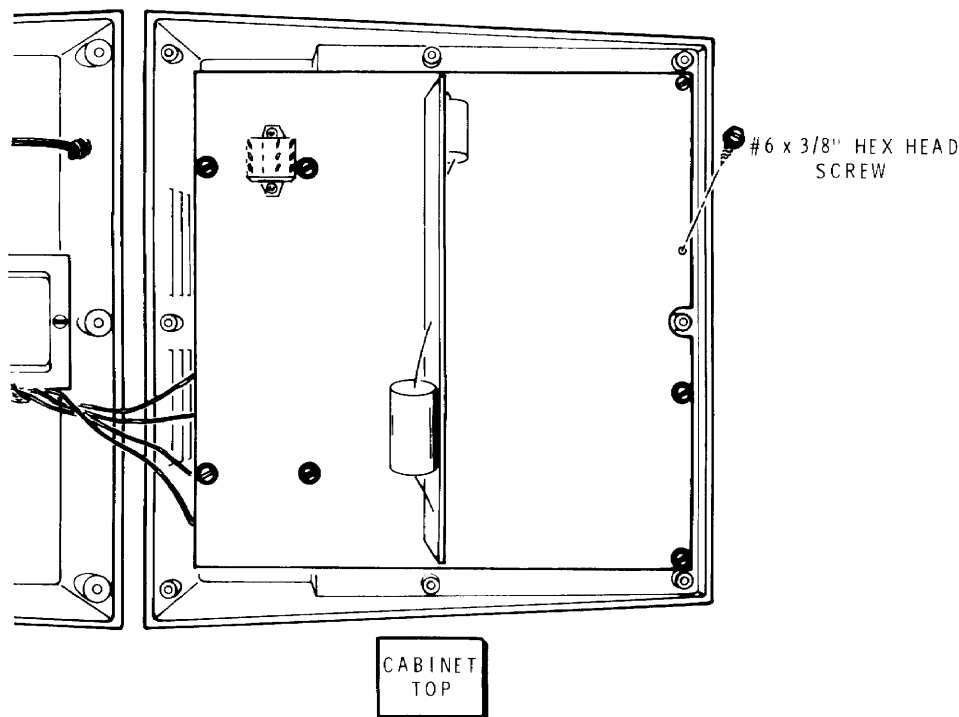
FINAL ASSEMBLY



PICTORIAL 7-1

Refer to Pictorial 7-1 for the following steps.

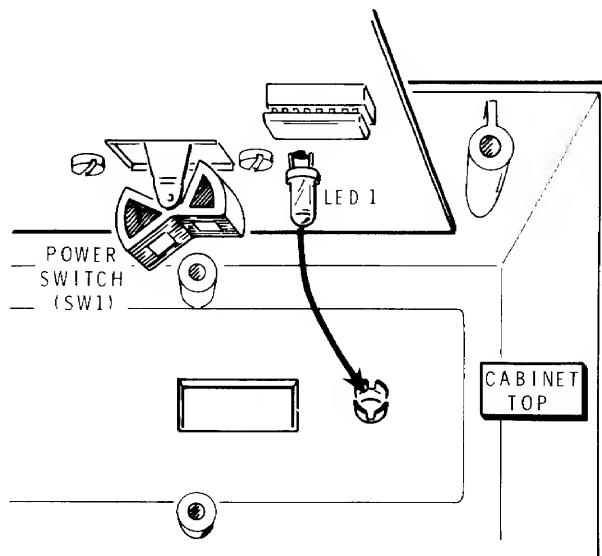
- () Remove the paper backing from the "Heathkit" label. Carefully press the label in place on the upper portion of the cabinet top as shown.
- () Press an LED grommet into the small round hole in the cabinet top.



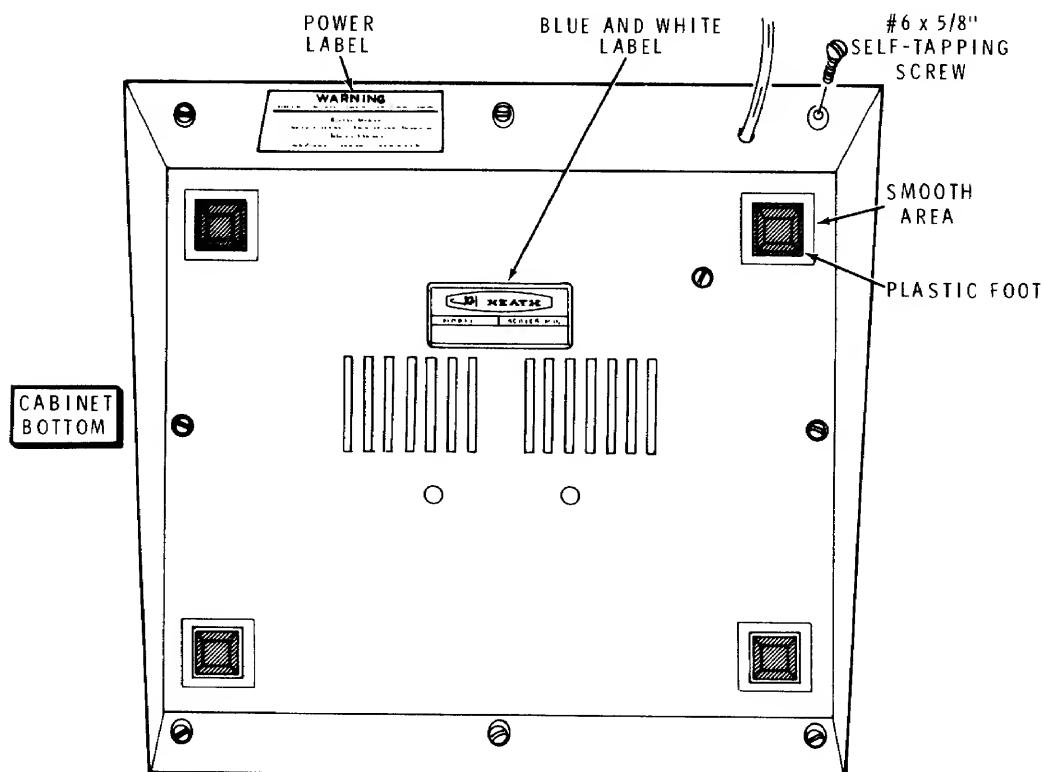
PICTORIAL 7-2

Refer to Pictorial 7-2 for the following steps.

- () Unplug the line cord.
- () Turn the cabinet top upside down and position it near the cabinet bottom as shown in the Pictorial. As you lower the main circuit board down onto the inverted cabinet top, be sure that LED1, next to the Power switch, fits straight down into the LED grommet as shown in Detail 7-2A. NOTE: If the LED protrudes through the cabinet top too far, resolder the LED's leads so the LED is closer to the circuit board.
- () Secure the main circuit board to the cabinet top with eight #6 \times 3/8" hex head screws.



Detail 7-2A



PICTORIAL 7-3

Refer to Pictorial 7-3 for the following steps.

- () Turn the cabinet top and main circuit board assembly right side up and fit the assembly into the cabinet bottom.
- () Turn both cabinet halves bottom-side up as shown in the Pictorial; then secure the bottom to the top with eight #6 × 5/8" self-tapping screws.
- () Remove the paper backing from the four feet and press them in place on the cabinet bottom in the smooth areas near the four corners as shown.

- () Remove the paper backing from the blue and white label and press the label in place on the cabinet bottom. NOTE: Be sure to refer to the numbers on the blue and white label in any correspondence you have with the Heath Company about your kit.
- () Remove the paper backing from the power label and press the label in place near the line cord as shown in the Pictorial.

This completes the "Final Assembly" of your kit. Proceed to "Operation."



OPERATION


This section of the Manual describes the operation of your Trainer, explains the keyboard commands, describes how to enter programs, has several sample programs, contains the monitor listing and several subroutine flowcharts, shows the memory map, and lists the entire 6800 instruction set.

Pictorial 8-1 (Illustration Booklet, Page 8) gives a brief description of the switches, LED's, and connectors.

KEYBOARD


The keyboard allows you to quickly enter commands and data to the microprocessor. After you press the RESET key, the display will show CPU UP, and the next keyboard entry will be interpreted as a command. The following paragraphs discuss the various commands.

Display Accumulator A

 Press this key and the contents of accumulator A will be displayed. The first four digits and decimal point identify the display, and the next two digits show the contents of the accumulator.

In the following example, the contents of accumulator A is $4A_{16}$ (or binary 01001010).


Example: Acca.4A

Now you may change the contents of accumulator A if you wish. To do this, press the  key. The display will now be:

Acca. _ _

With two key strokes, enter the new hexadecimal number you want in accumulator A.

Display Accumulator B


 Press this key and the contents of accumulator B will be displayed. A typical display is:

Accb.5F


In this example, accumulator B contains $5F_{16}$ (binary 01011111).

The contents of accumulator B can be changed in the same way that accumulator A is changed.


Display Program Counter

 Press this key and the contents of the microprocessor's program counter will be displayed. The first two digits and decimal point identify the display, and the next four digits show the contents of the program counter.


Example: Pc.0040

In this example, the program counter contains 0040_{16} . You may change the program counter by pressing the  key and then entering the new hexadecimal number.

Display Index Register

 Press this key and the contents of the index register will be displayed.

Example: In.FDF4.

You can change the register by pressing the  key and then entering a new hexadecimal number.

Display Condition Codes Register

5C Press this key and the contents of the condition codes register (1's and 0's) will be displayed. The display letters (H, I, N, Z, V, and C) correspond to the letters assigned to the six condition codes. (See the "instruction set" on Page 89.)

Example: 001001

This register **cannot** be changed by pressing the **CHAN C** key.

Display Stack Pointer Register

SP 6 Press this key and the contents of the stack pointer register will be displayed.

Example: SP.00d2

This register **cannot** be changed by pressing the **CHAN C** key.

Resume User's Program

RTI 7 Press this key and your program will start at the location contained in the program counter. This key is used to return to normal user program operation from breakpoints or single stepping.

Single Step User's Program

SS 8 Press this key and the microprocessor will perform only one step of your program. The instruction to be performed is taken from the address contained in the program counter. After the step, the next instruction and its address are displayed. The displayed instruction may be changed by pressing the **CHAN C** key and then entering the new data. Also at this time, you may examine registers, memory, or use any of the other monitor functions.

Set Breakpoint

BP 9 Press this key and you can then make an entry into the monitor breakpoint table. A breakpoint is a point where you want to stop the program to examine the microprocessor registers, memory, etc.

The display is _ _ _ _ br.

Enter the four digits of a hexadecimal address for the breakpoint. The address must be the address of an operational code in your program and that code must be in RAM. No breakpoints are possible in ROM. You may have up to four breakpoints in your program at any one time.

Do not press the RESET key. This clears all the breakpoints.

If you make an incorrect entry, and the entry is still displayed, press the **CHAN C** key as many times as necessary for the display to return to _ _ _ _ br. Then enter the correct address.

Auto Load Of Memory

AUTO A Press this key and _ _ _ _ Ad will be displayed.

Enter the address you want to start at. Example: Enter 0, 0, A, and 4. The display is now:

00A4 _ _ .

Enter the 2-digit hexadecimal value you want entered at that address.

The display will now advance to the next address. You can continue changing memory data until you press the RESET key.

Display Previous Address

BACK B Press this key when an address and its data are displayed (you are examining memory with the E function, your program has come to a breakpoint, or you are single stepping your program), and the previous address and its data will be displayed. You may change this data by pressing the **CHAN C** key and then entering the new data.



Change Displayed Value

CHAN C Press this key when an address and its data are displayed, and the data will be replaced with “_ _ _ _”. Then enter the new hexadecimal value you want at this address.

You may use this function to correct a value you entered by mistake. However, if the monitor is expecting a command and the change function is not valid, the change command will be ignored.

DO User Program

DO D Press this key and the display will become:

_ _ _ _ do.

Enter the beginning address of your program. Your program will now start at the new address instead of where the program counter was pointing. The display will become blank and the program will run until a display is called for, until it comes to a breakpoint, or until you press the reset key.

This key function combines several other functions. You could get the same result by displaying and changing the program counter and then pressing the **RTI 7** key.

Examine Memory

EXAM E Press this key and the display will become:

_ _ _ _ Ad.

Enter a new address. The display will now indicate the data at this new address. You may now change the displayed value by using the **CHAN C** key or you can step backwards or forwards through memory using the **BACK B** and **FWD F** keys.

Display Next Address

FWD F Press this key when an address and its data are displayed, and the next address and its data will be displayed. You may change this data by pressing the **CHAN C** key and then entering the new data.

ENTERING PROGRAMS

Pictorial 8-2 shows the first two instructions of Sample Program 1 (in the following section) and indicates the various information they contain. This information is further described in the following paragraphs.

Instruction Address: This is usually called the Program Counter. In order to perform an instruction, the Program Counter must contain the address that is in this column. RTI and SS require the Program Counter to contain the address that is in this column for proper execution. The address entered after DO is pressed must be an instruction address. Breakpoints are not recognized except at instruction addresses.

Instruction: This is one, two, or three bytes of data as required by the addressing mode used.

Op code: This is a "byte of information referred to as machine code, it indicates in hexadecimal the operation to be performed.

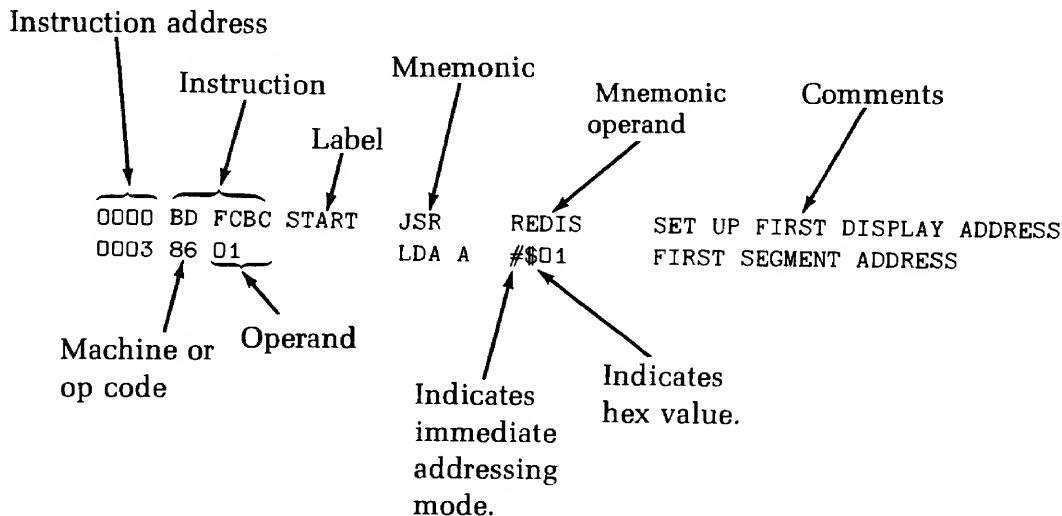
Operand: This is additional hexadecimal information required to perform the operation. It may be zero, one, or two bytes as determined by the addressing mode.

Label: This is usually a name applied to a subroutine in the program used more than once. In the sample programs, the address to be entered to begin execution is labeled "Start."

Mnemonic: This is a three-letter indication of the source instruction. A fourth letter, A or B, is added to indicate which of two accumulators if the instruction applied to either one.

Mnemonic operand: Again, this is additional information that is required for the operation. It may be a label, address, or data. The \$ sign indicates the information is a hex value. The # sign indicates the immediate addressing op code is to be used.

Comments: This is a brief description of what is happening in the program.



PICTORIAL 8-2



When you load a program into the Trainer, only the one, two, or three bytes of each instruction are entered. You may use either of two modes to enter the instructions: "Auto", or the more laborious "Examine and Change." Forward, Back, and Change are valid commands in the Examine, and Change mode and may be used to correct entry errors. However, they are not valid in Auto. If you make an error in the Auto mode, press the Reset and Auto keys. Then enter the address where the error was made and continue from there; or, remember where the error was made and then examine and change that memory location after you finish entering the entire program.

The following charts show the sequence of events to enter the first two instructions of sample program 1. The first chart shows the Examine and Change mode while the second chart shows the Auto mode.

Examine and Change

Press the EXAM key and then enter the first instruction address, 0000, by pressing the 0 key four times. Then check the display and continue to enter the program as shown below.

Display is	→	Press	→	Enter	→	Display is	→	Press
0000XX		CHAN		BD		0000BD		FWD
0001XX		CHAN		FC		0001FC		FWD
0002XX		CHAN		BC		0002BC		FWD
0003XX		CHAN		86		000386		FWD
0004XX		CHAN		01		000401		FWD
0005XX		CHAN		etc.				

Auto

Press the AUTO key and then enter the first instruction address, 0000, by pressing the 0 key four times.

Display is	→	Enter
0000--		BD
0001--		FC
0002--		BC
0003--		86
0004--		01
0005--		etc.

Press RESET after last entry to exit the AUTO mode.

If Examine and Change is used, the last entry in sample program 1 (Page 55 and last page of Illustration Booklet) results in the display 0025 DA, and this display remains until a new command is entered through the keyboard.

If Auto is used, the last display will be the address of the next continuous memory location which is the last program instruction address plus the number of bytes in the instruction. In this program, 0024 plus two; 0026_ _ . The dash (or "prompt" characters) are displayed in the data locations.

After you enter a program, by either method, check the ending address to be sure that you have not omitted or double entered data.

Enter sample program 1 (on Page 55 and last page of Illustration Booklet) into your Trainer. Use either of the two entry methods.

If you used the Examine and Change mode to enter the program, the program can be run by pressing DO and entering the address of the instruction labeled "Start," 0000. If you used the Auto mode, first press the RESET key to exit the Auto mode. Then press DO and enter the address of the instruction labeled "Start," 0000.

USING BREAKPOINTS

We will now use sample program 1 to show how programs can be inadvertently changed and even "crash" when breakpoints are inserted at improper locations (at addresses other than the instruction address).

Press RESET and insert breakpoints at 0004 and 0005.

Press BR 0004.

Press BR 0005.

Start the program by pressing DO 0000.

Notice that the CPU has ignored the breakpoint at 0004 and stopped at 0005.

Examine 0003 and 0004 by pressing EXAM, 0003, and FWD. The instructions there are correct (86 and 01).

Examine accumulator A by pressing the ACCA key.

Accumulator A has been loaded with the software interrupt instruction 3F that was temporarily placed at 0004 by the breakpoint at that address.

Watch the "H" display and press RTI. The 3F in accumulator A caused the first display to be incorrect. The program will stop again at 0005.

Insert a breakpoint at 0002 and then press RTI to resume execution. The program will run until it comes to 0002 which is changed by the breakpoint to 3F. The program will "crash" because of the wrong instruction and one of three things will happen: The display will be blank; all eights will be displayed; or all eights will appear, followed by CPU UP. In any case, press Reset to return control to the monitor program.

Press EXAM and enter 0000. Use FWD and CHAN to examine and correct errors introduced when the program crashed. You will always find the data at the breakpoint addresses has been changed. More often than not, the data at the breakpoint addresses will become 3F, although this may also change because the program crashed. Before you proceed, run the program to be sure all errors have been corrected.

In order to properly execute SS or RTI, the program counter must contain the instruction address where you wish to start. If single step begins at an incorrect address, the single step routine will not execute an invalid instruction and the display will not change. If the instruction at the PC address is a valid opcode, SS will execute the instruction using the following bytes as necessary and will continue unless it comes to an invalid instruction. RTI will try to execute the instruction in the same manner; except in the case of an invalid instruction, the program will probably crash. We will use SS to illustrate what happens.

Push RESET. Examine the Program Counter by pushing PC. Then change it to 0016 by pushing CHAN and 0016.

Press SS. The instruction at 0016 is not a valid instruction. In the single step mode, the machine will reject the instruction and 0016 FD will continue to be displayed and nothing happens. If RTI is pressed, the program will crash as it would when an invalid instruction is encountered. Probably only the first in-



struction will be changed, if any, in this particular circumstance. If you press RTI to see what happens, examine the program afterwards and correct any errors introduced; then run the program to be sure it is correct before proceeding.

Examine and change the program counter to 000F by pressing PC, CHAN, and 000F. Press SS. In this case FE is a valid instruction, LDX extended, and X is loaded from non-existent memory locations 3ACE and 3ACF.

Press SS. Here again 2F is a valid instruction (a conditional branch BLE). A branch may occur to 0015 or the program may fall through to 0014. In either case, two incorrect instructions have been performed in place of two or three correct instructions introducing error in the program. This is of no great consequence in this program but may be in another. Since an invalid instruction was not encountered, placing the program counter at 000F and pressing RTI would do exactly the same thing.

Now sample program 1 will be used to illustrate a procedure using breakpoints and single step to go through a program.

There are two important considerations pertaining to reserved memory bytes to keep in mind. First; DIGADD is used by all monitor routines. If you examine

these memory locations, 00F0 and 00F1, you will always find C12F, "V" display address, there because the examine command puts it there before it outputs the data. Secondly; DIGADD is always loaded with C16F, "H" display address, when DO or RTI are used.

Single step uses RMB TEMP, T1 and T0 in common with many of the monitor routines. Single step will replace information stored at these locations by the monitor routines. As a result, the routine may return with incorrect information or it may not be able to return at all and the program will crash.

When the program stops, at a breakpoint or after a single step, the address of the next instruction (contained in the program counter) and the instruction will be displayed. You may examine and make changes to any register (except stack pointer) or address provided you DO NOT change the program counter. The instruction displayed when the program stopped will be the next one executed when SS or RTI is pressed, regardless of what is being displayed.

The following procedure gives instructions. The six characters on the right, on the same line, indicate what the display should be after you perform the instruction. You will be instructed to examine registers affected by the instruction that has been executed.

You may examine any other registers or memory locations if you wish. The comment after an instruction is explanatory information.

<u>INSTRUCTION</u>	<u>DISPLAY</u>
Press RESET.	CPU UP.
Press PC, CHAN, and enter 0000. The program counter now contains the start address.	Pc 0000
Press SP. This is the next location available on the stack. The JSR instruction should store the address for return from the REDIS subroutine (0003) at this location.	SP 00d2
Press SS, Jump to REDIS.	FCbC dF
Press EXAM and 00D1.	00d1 00
Press FWD. Return address is on stack.	00d2 03
Press BR and enter 0003. To get past monitor routine.	0003 br
Press RTI. Might normally use examine to check result of routine. In this case, DIGADD RMB is loaded with C16F. Examine will just change what is there.	0003 86
Press SS.	0005 20
Press ACCA. (A) loaded with correct value.	Acca.01
Press SS. Branch to OUT-offset correct.	000E bd
Press SS. Jump to OUTCH.	FE3A dF
Press BR and enter 0011. To get past monitor routine. Could check stack here if desired.	0011 br
Press RTI. Exit OUTCH address of next display in DIGADD; Do not check.	0011 CE
Press SS.	0014 09
Press INDEX. Is (X) loaded?	In.2F00
Press SS.	0015 26
Press INDEX. Is (X) decremented?	IN.2EFF
Press CC. Z bit clear if (X) not 0000 yet.	XXX0 XX (X = don't care)
Press SS. Branches to WAIT if Z was clear.	0014 09
Press INDEX, CHAN, and enter 0001.	In.0001
Press SS.	0015 26



Press CC. (X) decremented to 0000 sets Z bit. Should drop through branch now.	XXX1 XX
Press SS. It did.	0017 16
Press ACCA.	Acca.01
Press SS.	0018 5d
Press ACCB. What was in (A) should be in (B).	Accb.01
Press SS.	0019 26
Press CC. Z bit clear if (B) not 00.	XXX0 XX
Press SS. Branches to SAME if Z is clear.	0007 d6
Press SS.	0009 Cb
Press ACCB. When the program runs normally, (B) at this point would be 5F because exit from OUTCH would be with the next display address, C15F, in DIGADD. Single step has caused DIGADD to be C10F.	Accb.0F
Press SS.	000b d7
Press ACCB. Hex 10 has been added to (B).	Accb.1F
Press SS. (B) has been stored at DIGADD. No reason to examine 00F1 since EXAM and SS will change what is there anyway.	000d 48
Press ACCA.	Acca.01
Press SS.	000E bd
Press ACCA. ACCA was 0000 0001 binary (01 hex). It has been shifted left and is now 0000 0010 binary (02 hex). The program is back to jump to OUTCH again. The same method as used before would get you back 0019 again. The program has proven good to that point so we will use a different method.	Acca.02
Press Reset. This clears the previous breakpoints.	CPU UP
Press BR and enter 0018.	0018 br.
Press DO and enter 0000. You may have noticed the program ran up to the breakpoint and the counter segment in "H" was momentarily lit. Now you are in another loop. You could press RT1 seven times and go back through the loop until (B) is 00. Again, since the branch is operating properly it is easier to change (B) to 00 and continue.	0018 5d
Press ACCB, CHAN, and enter 00.	Accb.00



Press SS.	0019 26
Press CC. The Z bit is set and the program should fall through the branch.	XXX1 XX
Press SS. It did.	001b 86
Press SS.	001d de
Press ACCA. (A) is loaded correctly.	Acca.01
Press SS.	001F 8C
Press Index. This is DIGADD again. Although the program has just finished with the "H" display, single step has placed C10F in DIGADD. This happens to be the address that will be in DIGADD after DP goes out in the "C" display and should result in a branch back to START.	In.C10F
Press SS. Same conditional BRANCH.	0022 26
Press CC. Z is set and the program should fall through.	XXX1 XX
Press SS. It did.	0024 20
Press SS. Every instruction in the program has been run except for the conditional branch at 0022.	0000 bd
Press Reset. Clears the breakpoint at 0018	CPU UP
Press BR and enter 001F.	001F br
Press DO and enter 0000.	001F 8C
Press Index. This time the program runs straight through until after (X) is loaded from DIGADD (at 001D) without an intervening single step or breakpoint. All segments were turned on and off in the "H" display and "I" display address C15F is in the index register as it should be.	In.C15F
Press SS. Conditional branch.	0022 26
Press CC. Z is clear and a branch to out should take place.	XXX0 XX
Press SS. It did.	000E bd
The entire program has now been run.	

SAMPLE PROGRAMS

These sample programs will give you practice entering programs and show the use of Monitor sub-routines.

SAMPLE 1

TURN ON AND OFF EACH SEGMENT IN
SEQUENCE BEGINNING AT H DISPLAY
USES MONITOR SUBROUTINES REDIS AND OUTCH
NOTE: ONE DP IN EACH DISPLAY IS ACTIVE

```

0000 BD FCBC START JSR REDIS SET UP FIRST DISPLAY ADDRESS
0003 86 01 LDA A #$01 FIRST SEGMENT CODE
0005 20 07 BRA OUT
0007 D6 F1 SAME LDA B DIGADD+1 FIX DISPLAY ADDRESS
0009 CB 10 ADD B #$10 FOR NEXT SEGMENT
000B D7 F1 STA B DIGADD+1
000D 48 ASL A NEXT SEGMENT CODE
000E BD FE3A OUT JSR OUTCH OUTPUT SEGMENT
0011 CE 2F00 LDX #$2F00 TIME TO WAIT
0014 09 WAIT DEX
0015 26 FD BNE WAIT TIME OUT YET?
0017 16 TAB
0018 5D TST B LAST SEGMENT THIS DISPLAY?
0019 26 EC BNE SAME NEXT SEGMENT
001B 86 01 LDA A #$01 RESET SEGMENT CODE
001D DE F0 LDX DIGADD NEXT DISPLAY
001F 8C C10F CPX #$C10F LAST DISPLAY YET?
0022 26 EA BNE OUT
0024 20 DA BRA START DO AGAIN

```

SAMPLE 2

TURNS ALL DISPLAYS OFF AND ON

DISPLAYS HEX VALUE AT 0044

USES MONITOR SUBROUTINES REDIS, OUTCH AND OUTHEX

```

0030 BD FCBC START JSR REDIS FIRST DISPLAY ADDRESS
0033 4F CLEAR CLR A
0034 BD FE3A JSR OUTCH TURN ALL SEGMENTS OFF
0037 DE F0 LDX DIGADD NEXT DISPLAY
0039 8C C10F CPX #$C10F LAST DISPLAY YET?
003C 26 F5 BNE CLEAR
003E 8D 13 BSR HOLD
0040 BD FCBC JSR REDIS FIRST DISPLAY ADDRESS
0043 86 08 LDA A #$08 HEX VALUE TO DISPLAY
0045 BD FE28 OUT JSR OUTHEX OUTPUT CHARACTER
0048 DE F0 LDX DIGADD NEXT DISPLAY
004A 8C C10F CPX #$C10F LAST DISPLAY YET?
004D 26 F6 BNE OUT
004F 8D 02 BSR HOLD
0051 20 DD BRA START DO AGAIN
0053 CE FF00 HOLD LDX #$FF00 TIME TO WAIT
0056 09 WAIT DEX
0057 26 FD BNE WAIT TIME OUT YET?
0059 39 RTS

```



SAMPLE 3

OUTPUTS MESSAGE BY DISPLAYING UP TO SIX
 CHARACTER WORD ONE WORD AT A TIME
 USES MONITOR SUB ROUTINE OUTSTO
 NOTE: DP MUST BE LIT TO INDICATE END OF STRING
 TO EXIT OUTSTR. DP IS PLACED IN THE
 SEVENTH DISPLAY POSITION TO FULFILL THIS
 REQUIREMENT WITHOUT ACTUALLY BEING DISPLAYED.

```

0060 BD FD8D START   JSR      OUTSTO  LEFT DISPLAY  OUT WORD
0063 00              FCB      $00,$3B,$7E,$3E,$05,$00,$80  YOUR
0064 3B
0065 7E
0066 3E
0067 05
0068 00
0069 80
006A 8D 3F          BSR      HOLD     HOLD DISPLAY
006C BD FD8D        JSR      OUTSTO  LEFT DISPLAY  OUT WORD
006F 00              FCB      $00,$79,$33,$7E,$7E,$00,$80  3400
0070 79
0071 33
0072 7E
0073 7E
0074 00
0075 80
0076 8D 33          BSR      HOLD     HOLD DISPLAY
0078 BD FD8D        JSR      OUTSTO  LEFT DISPLAY  OUT WORD
007B 00              FCB      $00,$00,$30,$5B,$00,$00,$80  IS
007C 00
007D 30
007E 5B
007F 00
0080 00
0081 80
0082 8D 27          BSR      HOLD     HOLD DISPLAY
0084 BD FD8D        JSR      OUTSTO  LEFT DISPLAY  OUT WORD
0087 00              FCB      $00,$00,$3E,$67,$00,$00,$80  UP
0088 00
0089 3E
008A 67
008B 00
008C 00
008D 80
008E 8D 1B          BSR      HOLD     HOLD DISPLAY
0090 BD FD8D        JSR      OUTSTO  LEFT DISPLAY  OUT WORD
0093 00              FCB      $00,$00,$7D,$15,$3D,$00,$80  AND
0094 00
0095 7D
0096 15
0097 3D
0098 00
0099 80
  
```



```
009A 8D 0F      BSR      HOLD      HOLD DISPLAY
009C BD FD8D    JSR      OUTST0    LEFT DISPLAY  OUT WORD
009F 05         FCB      $05,$1C,$15,$15,$10,$15,$80  RUNNIN
00A0 1C
00A1 15
00A2 15
00A3 10
00A4 15
00A5 80
00A6 8D 03      BSR      HOLD      HOLD DISPLAY
00A8 7E 0060    JMP      START     DO AGAIN
00AB CE FF00    LDX      #$FF00    TIME TO WAIT
00AE 09         DEX
00AF 26 FD      BNE      WAIT      TIME OUT YET?
00B1 39         RTS
```

SAMPLE 4
 OUTPUTS SAME MESSAGE AS PROGRAM 3
 IN TICKER TAPE FASHION
 USES MONITOR SUB ROUTINES REDIS AND OUTSTR

```

0000 7F 0007 START CLR MORE+1 CLEAR POINTER
0003 CE 002A NEXT LDX #MESSA MESSAGE ADDRESS
0006 A6 00 MORE LDA A 0,X GET CHARACTER
0008 A7 2D STA A OUT+3-MESSA,X STORE CHAR. AT OUT PLUS
000A 08 INX NEXT CHARACTER
000B 8C 0030 CPX #30 FULL STRING YET?
000E 26 F6 BNE MORE
0010 8D 11 BSR HOLD HOLD DISPLAY
0012 BD FCBC JSR REDIS FIRST CHAR. TO "H" DISPLAY
0015 BD 0054 JSR OUT
0018 96 07 LDA A MORE+1 FIRST CHARACTER NUMBER
001A 4C INC A MOVE STRING UP ONE CHARACTER
001B 97 07 STA A MORE+1 NEW FIRST CHARACTER
001D 81 25 CMP A #25 LAST CHARACTER TO "H" YET?
001F 26 E2 BNE NEXT BUILD NEXT STRING
0021 20 DD BRA START DO AGAIN
0023 CE 6000 HOLD LDX #6000 TIME TO WAIT
0026 09 WAIT DEX
0027 26 FD BNE WAIT TIME OUT YET?
0029 39 RTS
002A 08 MESSA FCB $08,$08,$08,$08,$08,$08 ----
002B 08
002C 08
002D 08
002E 08
002F 08
0030 3B FCB $3B,$7E,$3E,$05,$00,$00 YOUR
0031 7E
0032 3E
0033 05
0034 00
0035 00
0036 79 FCB $79,$33,$7E,$7E,$00,$00 3400
0037 33
0038 7E
0039 7E
003A 00
003B 00
003C 30 FCB $30,$5B,$00,$00,$3E,$67 IS UP
003D 5B
003E 00
003F 00
0040 3E
0041 67
0042 00 FCB $00,$00,$7D,$15,$3D,$00,$00 AND
0043 00
0044 7D
0045 15
  
```



```
0046 3D
0047 00
0048 00
0049 05          FCB      $05,$1C,$15,$15,$10,$15  RUNNIN
004A 1C
004B 15
004C 15
004D 10
004E 15
004F 08          FCB      $08,$08,$08,$08,$08  -----
0050 08
0051 08
0052 08
0053 08
0054 BD FE52 OUT  JSR      OUTSTR  OUTPUT CHARACTER STRING
                        OUTPUT STRING STORED HERE
0057 00          FCB      $00,$00,$00,$00,$00,$00,$80
0058 00
0059 00
005A 00
005B 00
005C 00
005D 80
005E 39
```

SAMPLE 5

THIS PROGRAM CONTINUOUSLY CHANGES THE HEX VALUE STORED AT KEY+1 UNTIL ANY HEX KEY IS DEPRESSED. THE RIGHT DP IS LIT TO INDICATE A VALUE HAS BEEN SET. THE USER THEN DEPRESSES THE VARIOUS HEX KEYS TO LOOK FOR THE SELECTED VALUE. THE RELATIONSHIP OF DEPRESSED TO CORRECT KEY IS MOMENTARILY DISPLAYED AS HI OR LO. DP AGAIN LIGHTS INDICATING TRY AGAIN. DEPRESSING THE CORRECT KEY DISPLAYS YES! WHICH REMAINS UNTIL ANY KEY IS DEPRESSED SETTING A NEW VALUE TO FIND. USES MONITOR SUB ROUTINES ENCODE, OUTSTO, INCH

```

0060 7F 0086 START CLR KEY+1 CLEAR KEY POINTER
0063 C6 20 ILL LDA B #$20 VIOLATION COUNT
0065 BD FDBB ILL1 JSR ENCODE WAIT FOR ILLEGAL INTERVAL
0068 25 F9 BCS ILL STILL LEGAL?
006A 5A DEC B
006B 26 F8 BNE ILL1 NOT A FELONY
006D C6 20 LEGAL LDA B #$20 TIME UNTIL PAROLE
006F 8D 38 BSR CODE CHANGE KEY TO FIND
0071 BD FDBB LEGAL1 JSR ENCODE SET KEY TO FIND
0074 24 F7 BCC LEGAL KEY TO FIND SET?
0076 5A DEC B
0077 26 F8 BNE LEGAL1 GOOD KEY?
0079 BD FD8D OUTDP JSR OUTSTO OUTPUT STRING
007C 00 FCB $00,$00,$00,$00,$00,$80 DP TO "C"
007D 00
007E 00
007F 00
0080 00
0081 80

* DP LIT FIND SELECTED KEY
0082 BD FDF4 JSR INCH LOOK FOR KEY
0085 C6 86 KEY LDA B #KEY+1 GET KEY VALUE
0087 11 CBA IS IT RIGHT KEY?
0088 27 14 BEQ YES IF CORRECT
008A 22 2A BHI HIGH IF GREATER THAN KEY+1 VALUE
008C BD FD8D JSR OUTSTO OUTPUT STRING
008F 00 FCB $00,$00,$00,$00,$0E,$7E,$80 LO
0090 00
0091 00
0092 00
0093 0E
0094 7E
0095 80
0096 CE 6000 HOLD LDX #$6000 TIME TO HOLD DISPLAY
0099 09 WAIT DEX
009A 26 FD BNE WAIT LONG ENOUGH YET?
009C 20 DB BRA OUTDP TRY AGAIN
009E BD FD8D YES JSR OUTSTO OUTPUT STRING

```

```

00A1 00          FCB      $00,$00,$3B,$4F,$5B,$A0  YES!
00A2 00
00A3 3B
00A4 4F
00A5 5B
00A6 A0
00A7 20 B7      BRA      START      DO AGAIN
00A9 96 86      CODE    LDA A      KEY+1    CURRENT KEY VALUE
00AB 4C          INC A          NEXT KEY
00AC 97 86      STA A      KEY+1    KEY TO FIND
00AE 81 10      CMP A      #$10    CAN'T BE GREATER THAN F
00B0 26 03      BNE          GOOD
00B2 7F 0086    CLR      KEY+1    MAKE IT 0
00B5 39          GOOD    RTS
00B6 BD FD8D    HIGH    JSR      OUTST0   OUTPUT STRING
00B9 37          FCB      $37,$30,$00,$00,$00,$00,$80  HI
00BA 30
00BB 00
00BC 00
00BD 00
00BE 00
00BF 80
00C0 7E 0096    JMP      HOLD

```


SAMPLE 6

THIS IS A TWELVE HOUR CLOCK PROGRAM
THE ACCURACY IS DEPENDENT UPON THE MPU CLOCK
FREQUENCY AND THE TIMING LOOP AT START.
CHANGING THE VALUE AT 0005/6 BY HEX 100
CHANGES THE ACCURACY APPROXIMATELY 1 SEC/MIN.
HOURS,MINUTE,SECOND RMB 0001/2/3 ARE LOADED
WITH THE STARTING TIME. THE FIRST DISPLAY
IS ONE SECOND AFTER START OF THE PROGRAM.
SECONDS WILL BE CONTENT OF SECOND RMB +1.
USES MONITOR SUB ROUTINES REDIS,DSPLAY.
NOTE:START THE PROGRAM AT 0004.

0001	00	HOURS	RMB	1	
0002	00	MINUTE	RMB	1	
0003	00	SECOND	RMB	1	
0004	CE B500	START	LDX	#\$B500	ADJUST FOR ACCURACY
0007	09	DELAY	DEX		
0008	26 FD		BNE	DELAY	WAIT ONE SECOND
000A	C6 60		LDA B	#\$60	SIXTY SECONDS,SIXTY MINUTES
000C	0D		SEC		ALWAYS INCREMENT SECONDS
000D	8D 0F		BSR	INCS	INCREMENT SECONDS
000F	8D 10		BSR	INCMH	INCREMENT MINUTES IF NEEDED
0011	C6 13		LDA B	#\$13	TWELVE HOUR CLOCK
0013	8D 0C		BSR	INCMH	INCREMENT HOURS IS NEEDED
0015	BD FCBC		JSR	REDIS	RESET DISPLAY ADDRESS
0018	C6 03		LDA B	#3	NUMBER OF BYTES TO DISPLAY
001A	8D 16		BSR	PRINT	DISPLAY HOURS,MINUTES,SECONDS
001C	20 E6		BRA	START	DO AGAIN
001E	CE 0003	INCS	LDX	#SECOND	POINT X AT TIME RMB
0021	A6 00	INCMH	LDA A	0,X	GET CURRENT TIME
0023	89 00		ADC A	#0	INCREMENT IF NECESSARY
0025	19		DAA		FIX TO DECIMAL
0026	11		CBA		TIME TO CLEAR?
0027	25 01		BCS	STORE	NO
0029	4F		CLR A		
002A	A7 00	STORE	STA A	0,X	STORE NEW TIME
002C	09		DEX		NEXT TIME RMB
002D	07		TPA		
002E	88 01		EOR A	#1	COMPLEMENT CARRY BIT
0030	06		TAP		
0031	39		RTS		
0032	96 01	PRINT	LDA A	\$01	WHAT'S IN HOURS?
0034	26 03		BNE	CONTIN	IF NOT ZERO
0036	7C 0001		INC	HOURS	MAKE HOURS ONE
0039	08	CONTIN	INX		POINT X AT HOURS
003A	7E FD7B		JMP	DSPLAY	OUTPUT TO DISPLAYS

SAMPLE 7

THIS PROGRAM CALCULATES THE OP CODE VALUE FOR BRANCH INSTRUCTIONS USING THE LAST TWO DIGITS OF THE BRANCH AND DESTINATION ADDRESSES. THE BRANCH ADDRESS IS ENTERED FIRST AND DISPLAYED AT "H" AND "I". THE DESTINATION ADDRESS IS THEN ENTERED AND DISPLAYED AT "N" AND "Z". THE OP CODE IS THEN CALCULATED AND DISPLAYED AT "V" AND "C". THE DISPLAY IS HELD UNTIL NEW INFORMATION IS ENTERED. SINCE ONLY TWO DIGITS ARE ENTERED, IT IS NECESSARY TO MAKE AN ADJUSTMENT IF THE HUNDREDS DIGIT IN THE TWO ADDRESSES IS NOT THE SAME. FOR EXAMPLE TO CALCULATE THE OFFSET OF A BRANCH FROM 00CD TO 011B. SUBTRACT A NUMBER FROM BOTH ADDRESSES THAT WILL MAKE THE GREATER ADDRESS LESS THAN 100. FOR EASE OF CALCULATION IN THIS CASE, SUBTRACT C0 FROM BOTH ADDRESSES AND ENTER THE RESULTS 0D AND 5B IN THE PROGRAM. SINCE THE DIFFERENCE BETWEEN THE ADDRESSES IS UNCHANGED THE CORRECT OP CODE (4C) WILL BE DISPLAYED. IF THE DISTANCE IS TOO GREAT FOR BRANCHING NO. WILL APPEAR AT "V" AND "C". USES MONITOR SUB ROUTINES
REDIS IHB OUTBYT OUTSTR

0000 BD FCBC START	JSR	REDIS	FIRST DISPLAY AT "H"
0003 BD FE09	JSR	IHB	INPUT BRANCH ADDRESS
0006 16	TAB		PUT IT IN B
0007 BD FE09	JSR	IHB	INPUT DESTINATION ADDRESS
000A 11	CBA		FORWARD OR BACK?
000B 25 0C	BCS	BACK	IF BACK
000D CB 02 FRWD	ADD B	#\$02	ADJUST 2 BYTES
000F 10	SBA		FIND DISTANCE
0010 81 80	CMP A	#\$80	IS IT LEGAL?
0012 24 12	BCC	NO	IF NOT
0014 BD FE20 OUT	JSR	OUTBYT	OUTPUT BRANCH OP CODE
0017 20 E7	BRA	START	LOOK FOR NEW ENTRY
0019 40 BACK	NEG A		MAKE A MINUS
001A 1B	ABA		ADD A AND B
001B 8B 02	ADD A	#\$02	ADJUST 2 BYTES
001D 43	COM A		GET COMPLIMENT
001E 8B 01	ADD A	#\$01	MAKE IT TWO'S
0020 81 80	CMP A	#\$80	IS IT LEGAL?
0022 25 02	BCS	NO	IF NOT
0024 20 EE	BRA	OUT	OUTPUT BRANCH OP CODE
0026 BD FE52 NO	JSR	OUTSTR	OUTPUT STRING
0029 15	FCB	\$15,\$9D	NO.
002A 9D			
002B 20 D3	BRA	START	LOOK FOR NEW ENTRY

SUBROUTINE FLOW CHARTS

Following, are flow charts of several subroutines. These are helpful when you write your own programs. The entry requirements necessary to call these subroutines and their exit conditions are also shown.

RESET/MAIN Routine

When the Reset key is released, the CPU outputs FFFE and FFFF to get a starting address. This is the address of the top two locations in the monitor ROM which in turn outputs FC00, the beginning address of the reset routine.

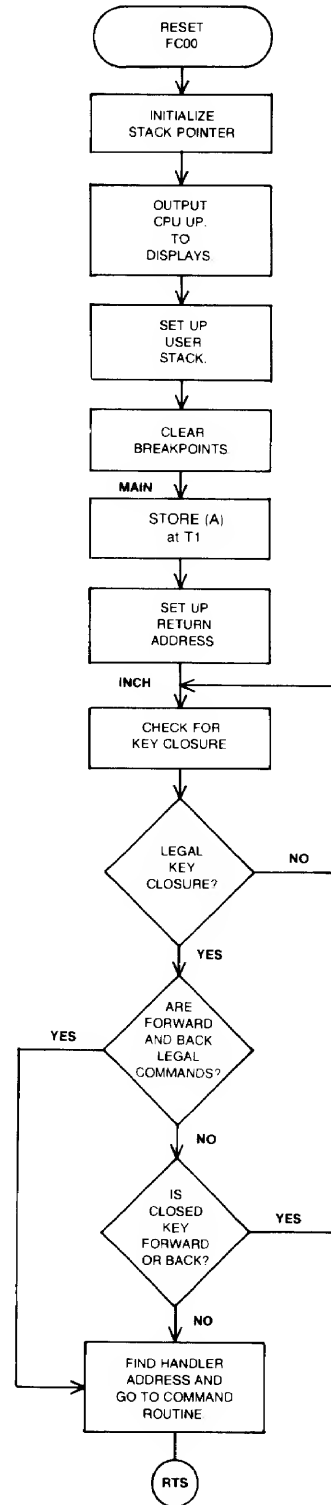
Reset first initializes the stack pointer to 00EB and outputs CPU UP, to the displays. The index register is set to 00CB (the start of the user's stack) and this value is stored in the user's stack pointer at location 00F2.

Breakpoints are cleared by placing FF in the eight RAM locations, 00E4-00EB. The program then goes into the main monitor loop. The contents of accumulator A, which is FF at this point, is stored at T1 and the address to return from command handler subroutines (FC19) is placed on the stack.

The program next calls INCH to scan and encode the keyboard. The program stays in INCH until a key is found closed.

The FORWARD and BACK commands are legal only after execution of the EXAM or SINGLE STEP commands. RAM location 00EE (T1) is cleared if FORWARD and BACK are legal commands. When INCH returns a key closure, T1 is tested to see if FORWARD and BACK are legal. If they are legal, a branch is made to MAIN 2 to obtain the subroutine address to handle the command and then goes to that handler. If FORWARD and BACK are not legal commands, tests are made to see that they are not the key closed before going to MAIN 2. If FORWARD or BACK is found to be the key closed, a branch back to MAIN 1 occurs and INCH is again called to look for a legal key closure.

RESET/MAIN ROUTINE



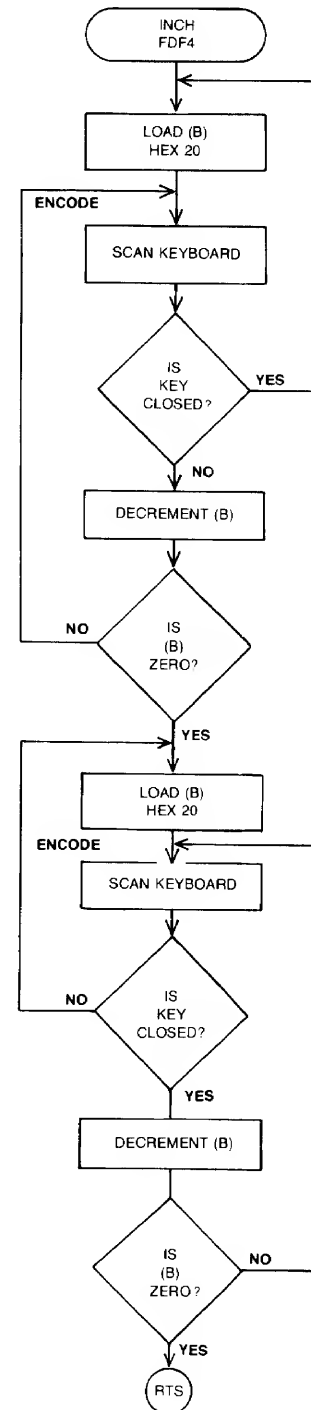
INCH Routine

INCH guards against the entry of a false output from the keyboard due to contact bounce or pressing more than one key. ACCB is loaded with hex 20 and ENCODE is called to scan the keyboard. If C is set (key closed), a branch occurs back to the beginning. If C is clear (no closure), ACCB is decremented and ENCODE is called again. ENCODE must return C clear 32 consecutive times (approx. 9 ms) to exit this loop. The second half of the routine is then entered. This half is identical to that described above, except C must be set 32 consecutive times before exit with the hex value of the key closed in ACCA.

INCH ROUTINE

ENTRY: None.

EXIT: (A) contains hex value of closed key.



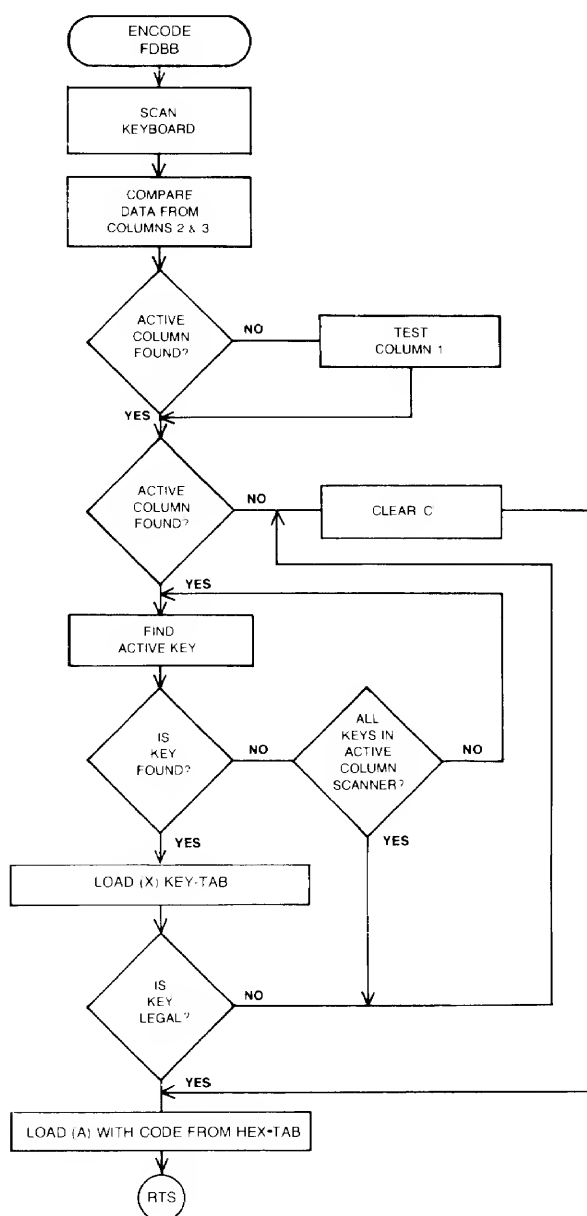
ENCODE Routine

ENCODE is the keyboard scanning routine. If a key is closed, the value is found in the hex table and loaded in ACCA. The C bit in the condition code register is set to indicate a valid key. If no key is closed or if the value is not in HEX-TAB, the C bit is cleared.

ENTRY: None.

EXIT: (A) contains hex value of key closed.
"C" set for valid condition.

ENCODE ROUTINE

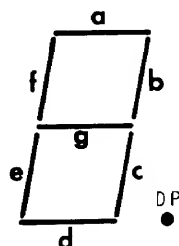


OUTCH Routine

OUTCH outputs a character to the display whose address is contained at memory location DIGADD (00F0-00F1). This routine may be entered at OUTO if the index register does not need to be saved. The code for the character to be displayed must be in accumulator A when the routine is entered. The following drawing shows the segment identification and the corresponding positions in the eight bits of accumulator A. A logic one in a bit will cause that segment to light, whereas a logic zero will keep it off. The hex and corresponding bit codes are shown for two characters used in the monitor program. The most significant bit is DP and the least significant bit is segment g.

Segment codes used by the monitor program are shown at the end of the monitor listing.

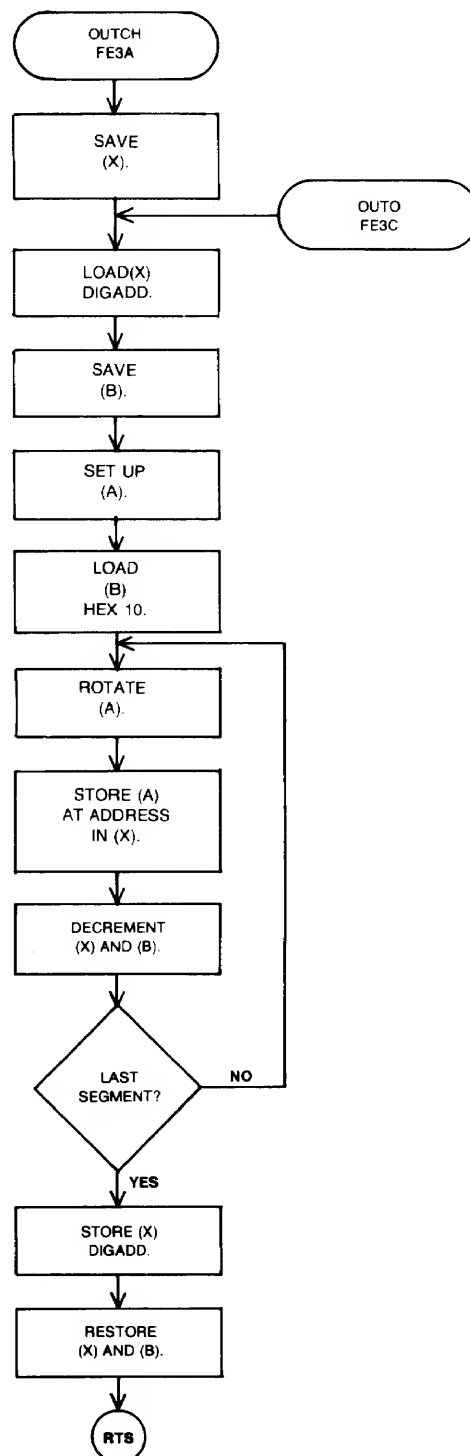
	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SEGMENT	DP	a	b	c	d	e	f	g
HEX C 4E	0	1	0	0	1	1	1	0
LTR c OD	0	0	0	0	1	1	0	1



OUTCH ROUTINE

ENTRY: ACCA contains segment code. DIGADD contains address of desired digit. Entry at OUTO if index register is to be saved.

EXIT: DIGADD contains address of next digit to right.





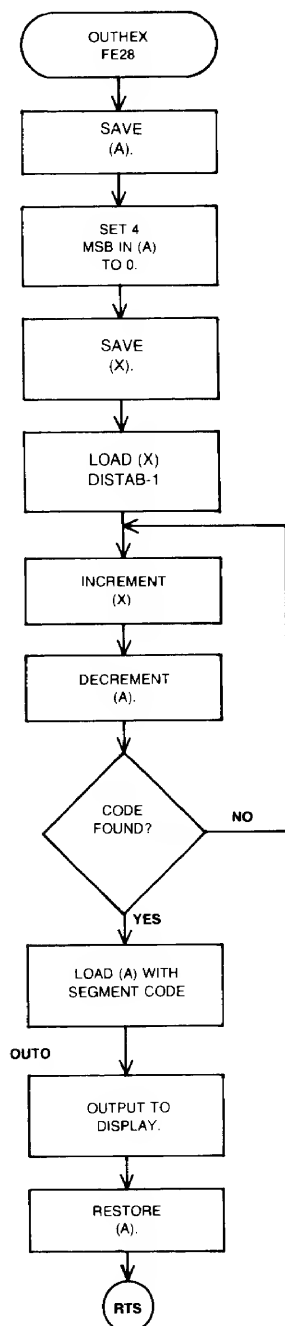
OUTHEX Routine

OUTHEX determines the segment code for a hex value contained in the four least-significant bits (LSB) of accumulator A. Subroutine OUTO is then called to output the hex value to the display whose address is obtained from DIGADD.

OUTHEX ROUTINE

ENTRY: ACCA contains hex value.

EXIT: Address of next digit to right contained in DIGADD.



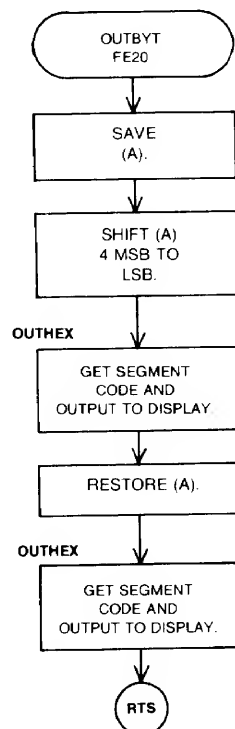
OUTBYT Routine

OUTBYT outputs two hex values contained in accumulator A to two adjacent displays. The value contained in the four most-significant bits (MSB) are moved to the LSB positions. OUTHEX is called to determine the segment code and in turn calls OUTO to output the character to the display addressed at DIGADD. Accumulator A is restored, and OUTHEX and OUTO are called again to output the LSB to the next display to the right.

OUTBYT ROUTINE

ENTRY: ACCA contains two hex values.

EXIT: Digit address for next digit to right contained in DIGADD.

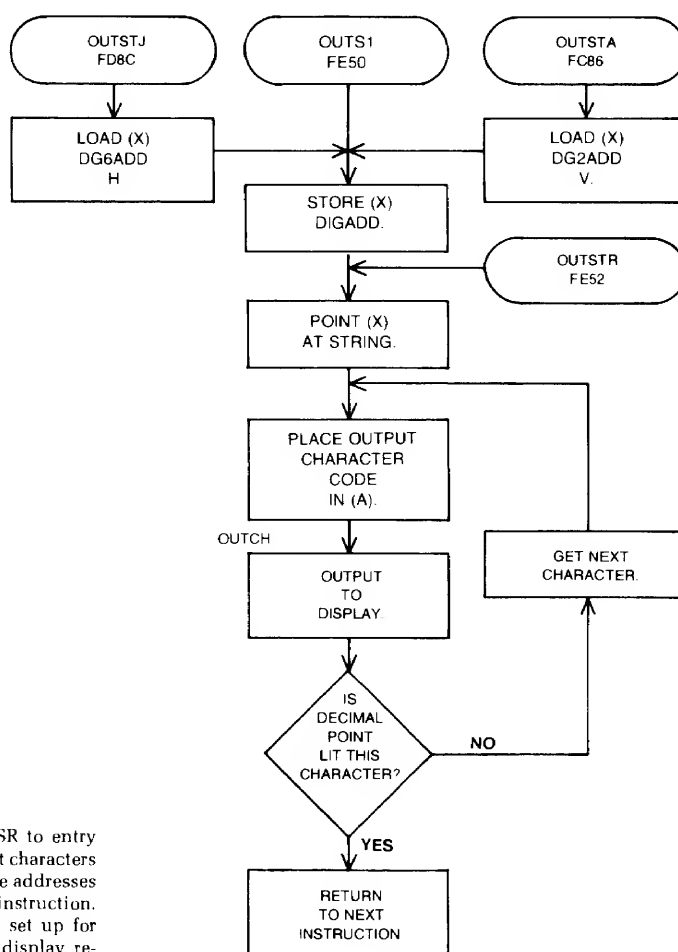


OUTST1 Routine

OUTST1 outputs a string of characters from left to right on the displays. The first character is output to the display whose address is contained in the index register upon entry to the routine. The last character must have the decimal point lit to indicate the end of the string. Adding hex 80 to the desired segment code causes the decimal point to be lit. For example, if the last character is to be LTR P, hex 67 (the last character code) would be hex 67 plus hex 80, or hex E7.

The routine may be entered at OUTSTJ or at OUTSTO (with the first character appearing in the left-most display) or at OUTSTA (with the first character appearing in the V display). Entry at OUTSTR requires the address for the first character to be in DIGADD. Exit from the routine is to the next instruction, which is one plus the address of the last character.

OUTST1 ROUTINE



ENTRY: Calling convention must be JSR to entry point. Segment codes for output characters from left to right at consecutive addresses immediately following jump instruction. Entry at OUTSTJ or OUTSTA set up for left-most character at H or V display respectively. Entry at OUTST1 requires (X) contain left-most digit address. Entry at OUTSTR requires left-most digit address at DIGADD. Decimal point must be lit on last character.

EXIT: To next instruction at 1 + address of last character.
ACCA is clear.
DIGADD contains address of display to right of last digit lit.

DSPLAY Routine

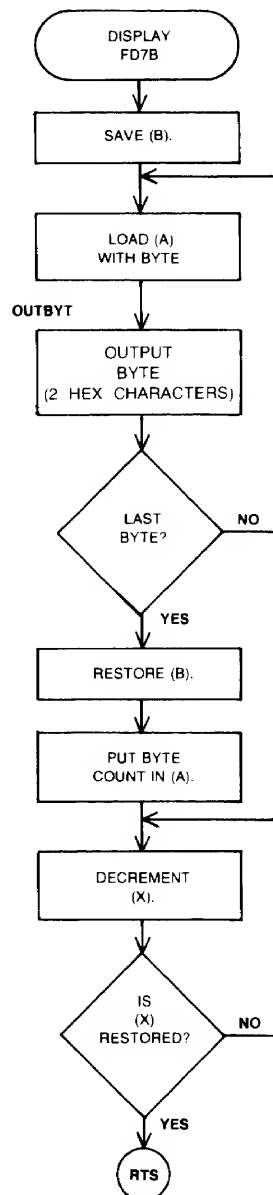
DSPLAY is called to output two or three bytes to the displays. The number of bytes to output is indicated by the contents of accumulator B. This routine could be called to output one byte, although OUTBYT would normally be called for this purpose.

Accumulator A is loaded with a byte value from an address contained in the index register and OUTBYT is called to output the byte to the displays. Then the index register is incremented to get the next byte, accumulator B is decremented, and OUTBYT is called again. When accumulator B is zero, all bytes have been output and the index register and accumulator B are restored before returning from the routine.

DSPLAY ROUTINE

ENTRY: (X) contains address of first byte.
(B) contains number of byte to output.
DIGADD contains address of digit.

EXIT: (X) and (B) unchanged.
Address of next digit to right contained in DIGADD.



IHB Routine

IHB outputs two hex characters to the displays corresponding to two consecutive key closures and returns to the calling routine with the byte value of the two closures in accumulator A.

INCH is called to get the value of the first key closure. OUTHEX is called to display the value on the display whose address is contained at DIGADD. The value contained in the four LSB of accumulator A is moved to the four MSB of accumulator A and then saved in accumulator B.

INCH is called again to get the value of the second key closure. OUTHEX is then called again and this value is displayed on the next display to the right.

The contents of accumulators A and B are combined and placed in accumulator A. Accumulator A now contains the byte value of the two closures. The MSB contains the first closure value and the LSB contains the second value. Accumulator B is restored, accumulator A is pushed onto the stack, and ENCODE is called to wait for the release of the second key. When the key is released, the byte is pulled from the stack and the program returns to the calling routine with the byte contained in accumulator A.

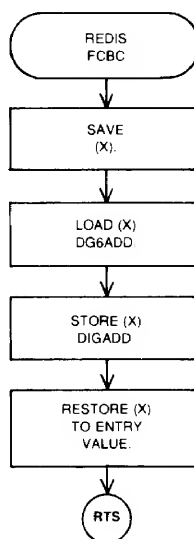
REDIS Routine

REDIS is a short routine to reset the address at DIGADD to the left-most display digit.

REDIS ROUTINE

ENTRY: None.

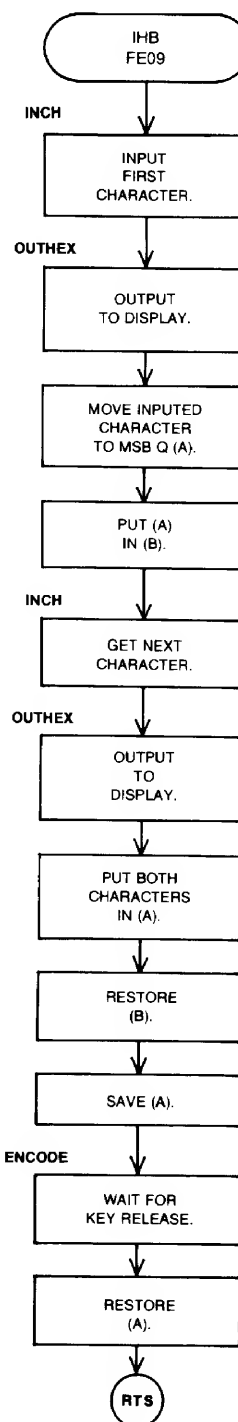
EXIT: DIGADD contains address of left-most digit.



IHB ROUTINE

ENTRY: None.

EXIT: ACCA contains byte value.
Digit address for next digit to right contained in DIGADD.





MONITOR LISTING

Your Trainer is controlled by IC12, the "read only memory" (ROM). The following is a listing of the program stored in this IC.

Tables at the end of the listing show labels used in the program, keyboard and display addresses, segment codes for characters displayed by the program, and addresses in RAM that are reserved for use by the monitor program.

```

FC00                                ORG    $FC00

                                **      RESET - CLEAR BREAKPOINT TABLE AND INITIALIZE STACK

FC00  8E 00 EB  RESET  LDS      #2*NBR+BKTBL-1
FC03  BD FD 8D          JSR      OUTST0
FC06  4E 67 3E          FCC      HEXC,LTRP,LTRU,0,LTRU,LTRP+$80
FC0C  CE 00 CB          LDX      #USRSTK
FC0F  DF F2            STX      USERS          SET UP FOR USER
FC11  86 FF            LDA      #$FF
FC13  C6 08            LDA      B      #2*NBR
FC15  36              RESE1  PSH      A
FC16  5A              DEC      B
FC17  26 FC            BNE      RESE1

                                **      MAIN - MAIN MONITOR LOOP
                                *
                                *      HANDLERS RETURN:
                                *      (B) = NUMBER BYTES SUBJECT TO "CHANGE"
                                *      (X) = ADDRESS BYTES SUBJECT TO "CHANGE"
                                *      (A) = 0 ENABLES "FORWARD" AND "BACK"

FC19  97 EE          MAIN  STA      A      T1
FC1B  86 19          LDA      A      #-MAIN/256*256+MAIN  LO ORDER RET.ADDR.
FC1D  36              PSH      A
FC1E  86 FC          LDA      A      #MAIN/256          HI ORDER BYTE RET. ADDR.
FC20  36              PSH      A          RETURN ONTO STACK
FC21  BD FD F4  MAIN1. JSR      INCH          GET COMMAND
FC24  7D 00 EE          TST      T1
FC27  27 08          BEQ      MAIN2          FORWARD OR BACK OK
FC29  81 0F          CMP      A      #$F
FC2B  27 F4          BEQ      MAIN1          ILLEGAL NOW
FC2D  81 0B          CMP      A      #$B
FC2F  27 F0          BEQ      MAIN1          ALSO ILLEGAL NOW
FC31  DF EC          MAIN2 STX      T0
FC33  CE FF B4          LDX      #CMDTAB-2
FC36  08          MAIN3  INX
FC37  08          INX          GET HANDLER ADDRESS
FC38  4A          DEC      A
FC39  2A FB          BPL      MAIN3
FC3B  A6 01          LDA      A      1,X          TARGET ADDRESS ONTO STACK
FC3D  36          PSH      A
FC3E  A6 00          LDA      A      0,X
FC40  36          PSH      A
FC41  DE EC          LDX      T0          RESTORE X
FC43  96 EE          LDA      A      T1
FC45  39          ZERO   RTS          JUMP TO HANDLER

                                **      ADDR - ACCEPT ADDRESS VALUE WITH 'AD' PROMPT
                                *
                                *      ENTRY, EXIT -- SEE 'DOPMT'

FC7E  DF EE          ADDR  STX      T1
FC80  8D 04          BSR      OUTSTA
FC82  77 BD          FCC      HEXA,LTRD+$80
FC84  20 EF          BRA      DOPM1

```



```

**      OUTSTA - OUTPUT STRING FOR ADDRESS PROMPT
*
FC86  CE C1 2F  OUTSTA  LDX      #DG2ADD
FC89  7E FE 50      JMP      OUTST1

**      DO - RESET USER PC AND RESUME
*
*      ENTRY:  NONE
*      EXIT:   TO 'RESUME'
*      USES:   ALL

FC8C  DE F2      DO      LDX      USERS
FC8E  08          INX
FC8F  08          INX
FC90  08          INX
FC91  08          INX
FC92  08          INX
FC93  08          INX          X TO USER PC
FC94  8D D9      BSR      DOPMT

**      RESUME - RESUME USER PROGRAM
*
*      1) BLANKS ALL DISPLAYS
*      2) INITIALIZES (DIGADD)
*      3) STEPS USER CODE PAST BREAKPOINT
*      4) INSERTS BREAKPOINTS
*      5) PRINTS INSTRUCTION UPON RETURN
*      ENTRY:NONE
*      EXIT: (B) = 1
*            (X) = USERPC
*      USES  ALL,T0,T1

FC96  8D 24      RESUME  BSR      REDIS          RESET DISPLAY
FC98  4F          CLR A
FC99  C6 06      LDA B  #6
FC9B  BD FE 3A  RES1  JSR      OUTCH          CLEAR DISPLAYS
FC9E  5A          DEC B
FC9F  26 FA      BNE      RES1
FCA1  8D 19      BSR      REDIS          RESET DISPLAY
FCA3  BD FE 6B  RES2  JSR      SSTEP          STEP PAST BREAKPOINT
FCA6  C6 04      LDA B  #NBR          SET BREAKPOINTS
FCA8  30          RES3  TSX
FCA9  EE 08      LDX      2*NBR,X          GET BREAKPOINT ADDRESS
FCAB  A6 00      LDA A  0,X
FCAD  36          PSH A
FCAE  36          PSH A
FCAF  86 3F      LDA A  #3F          REPLACE WITH SWI
FCB1  A7 00      STA A  0,X
FCB3  5A          DEC B
FCB4  26 F2      BNE      RES3
FCB6  CE FC CE  LDX      #BKPT
FCB9  7E FE FC  JMP      SWIVE1          GO TO USER CODE

**      REDIS - RESET DISPLAYS
*
*      ENTRY:  NONE
*      EXIT:   DIGADD SET TO LEFTMOST DIGIT
*      USES:   T0

FCBC  DF EC      REDIS  STX      T0
FCBE  CE C1 6F  LDX      #DG6ADD
FCC1  DF F0      STX      DIGADD
FCC3  DE EC      LDX      T0
FCC5  39          RTS

```



```

**      BADDR - BUILD ADDRESS
*
*      ENTRY: NONE
*      EXIT   (X) = ADDRESS

FCC6  CE 00 EE  BADDR  LDX    #T1
FCC9  8D B3      BSR     ADDR
FCCB  DE EE      LDX     T1
FCCD  39          RTS

**      BKPT - BREAK POINT RETURN
*      1) REMOVE BKPTS FROM USER CODE
*      2) CHECK FOR BREAKPOINT HIT AND EITHER
*          A) RESUME IF NO HIT
*          B) PRINT INSTRUCTION AND RETURN IF HIT

FCEE  30      BKPT  TSX
FCCF  9F F2      STS    USERS
FCD1  A6 06      LDA A   6,X
FCD3  26 02      BNE    BKP1      DECREMENT PC ON USERS STACK
FCD5  6A 05      DEC     5,X
FCD7  4A      BKP1  DEC A
FCD8  A7 06      STA A   6,X
FCDA  E6 05      LDA B   5,X
FCDC  D7 EC      STA B   T0      SAVE FOR COMPARE
FCDE  97 ED      STA A   T0+1

**      NOW CLEAR BREAKPOINTS

FCE0  0C      CLC              'C' IS HIT FLAG
FCE1  8E 00 D9  BKP2  LDS     #BKPTL-3-NBR-NBR
FCE4  C6 04      LDA B   #NBR
FCE6  32      BKP3  PUL A
FCE7  32      PUL A              OLD OP CODE INTO A
FCE8  30      TSX
FCE9  EE 08      LDX     2*NBR,X
FCEB  9C EC      CPX     T0      DO WE HAVE A HIT?
FCED  26 01      BNE    BKP4      NO WE DO NOT
FCEF  0D      SEC              YES WE DO - SET FLAG
FCF0      BKP4  EQU     *
FCF0  A7 00      STA A   0,X      FIX USER CODE
FCF2  5A      DEC B
FCF3  26 F1      BNE    BKP3
FCF5  24 AC      BCC    RES2      BREAKPOINT NOT HIT
FCF7  DE EC      LDX     T0      (X) = USER PC

**      MEM - DISPLAY ADDRESS AND DATA
*
*      ENTRY: (X) = ADDRESS
*      EXIT: (B) = 1
*      USES: A,B,C,T0,T1

FCF9  8D C1      MEM  BSR     REDIS      RESET DISPLAY
FCFB  DF EE      STX     T1
FCFD  CE 00 EE      LDX     #T1
FD00  C6 02      LDA B   #2
FD02  8D 03      BSR     MEM2      DISPLAY ADDRESS
FD04  EE 00      LDX     0,X
FD06  5A      DEC B
FD07  7E FD 7B  MEM2  JMP     DSPLAY      OUTPUT DATA

**      AUTO - AUTO LOAD OF MEMORY
*
*      ENTRY: NONE
*      EXIT: NO EXIT POSSIBLE
*      USES: ALL,T0,T1

```

```

FD0A 8D BA    AUTO    BSR    BADDR    BUILD ADDRESS
FD0C 8D EB    AUT1    BSR    MEM
FD0E 8D 0B    BSR    REPLAC
FD10 08      INX
FD11 20 F9    BRA     AUT1    NO EXIT

```

```

**      EXAM - EXAMINE MEMORY
*
*      ENTRY:  NONE
*      EXIT:   (X) = ADDRESS
*              (B) = 0
*              (A) = 0
*      USES:   ALL,T0,T1

```

```

FD13 8D B1    EXAM    BSR    BADDR    BUILD ADDRESS
FD15 09      DEX

```

```

**      FOWD - DISPLAY NEXT BYTE
*
*      ENTRY:  (X) = OLD ADDRESS
*      EXIT:   (X) = (XOLD) + 1
*              (B) = 1
*              (A) = 0
*      USES:   ALL,T0

```

```

FD16 08      FOWD    INX
FD17 08      INX

```

```

**      BACK - DISPLAY PREVIOUS BYTE
*
*      ENTRY:  (X) = ADDRESS
*      EXIT:   (X) = (XOLD) + 1
*              (B) = 1
*              (A) = 0
*      USES:   ALL,T0

```

```

FD18 09      BACK    DEX
FD19 20 DE    BRA     MEM    DISPLAY ADDRESS AND DATA

```

```

**      REPLAC - REPLACE DISPLAYED VALUE
*
*      'REPLAC' 1) BACKSPACES DISPLAY TO CANCEL DISPLAYED VALUE
*               2) SENDS PROMPT FOR REPLACEMENT VALUE
*               3) ACCEPTS AND REPLACES DESIGNATED BYTE(S)
*      ENTRY:  (X) = ADDRESS OF BYTE(S) TO REPLACE
*              (B) = NUMBER OF BYTES
*              (DIGADD) = ADDRESS OF DIGIT TO RIGHT OF DISPLAYED
*      EXIT:   B,X,DIGADD UNCHANGED
*      USES:   T0,A,C

```

```

FD1B 5D      REPLAC  TST B
FD1C 27 06    BEQ     REPL1    NO BYTES
FD1E 36      PSH A
FD1F 8D 22    BSR     BKSP     BACKSPACE DISPLAYS
FD21 8D 02    BSR     PROMPT
FD23 32      PUL A
FD24 39      REPL1  RTS

```

```

**      PROMPT - PROMPT AND INPUT BYTES
*
*      ENTRY:  (X) = ADDRESS TO STORE VALUE
*              (B) = NUMBER OF BYTES
*              (DIGADD) = ADDRESS OF FIRST ECHO CHARACTER
*      EXIT:   B,X UNCHANGED
*              DIGADD UPDATED
*      USES:   T0, DIGADD

```



FD25	37	PROMPT	PSH B		
FD26	86 08		LDA A	#DASH	PROMPT CHARACTER
FD28	58		ASL B		
FD29	BD FE 3A	PROM1	JSR	OUTCH	SEND PROMPT
FD2C	5A		DEC B		
FD2D	26 FA		BNE	PROM1	
FD2F	33		PUL B		
FD30	8D 11		BSR	BKSP	BACKSPACE DISPLAYS
FD32	37		PSH B		**ALTERNATE ENTRY**
FD33	BD FE 09	PROM2	JSR	IHB	GET BYTE VALUE
FD36	A7 00		STA A	0,X	PLACE INTO MEMORY
FD38	08		INX		RUMP POINTER
FD39	5A		DEC B		
FD3A	26 F7		BNE	PROM2	MORE TO GO
FD3C	33		PUL B		
FD3D	17		TBA		DUPLICATE
FD3E	09	PROM3	DEX		FIX X
FD3F	4A		DEC A		
FD40	26 FC		BNE	PROM3	
FD42	39		RTS		EXIT
** BKSP - BACKSPACE DISPLAYS					
*					
* ENTRY: (B) = NUMBER DIGIT PAIRS TO BACKSPACE					
* EXIT: (DIGADD) = (DIGADD) + 20 * (B)					
* USES: A,C					
FD43	37	BKSP	PSH B		
FD44	96 F1		LDA A	DIGADD+1	L.S. BYTE
FD46	8B 20	BKSP1	ADD A	#20	BACKSPACE TWO PLACES
FD48	5A		DEC B		
FD49	26 FB		BNE	BKSP1	
FD4B	97 F1		STA A	DIGADD+1	
FD4D	33		PUL B		
FD4E	39		RTS		
** REGISTER DISPLAY FUNCTIONS					
*					
* ENTRY: NONE					
* EXIT: (B) = NUMBER BYTES THIS REGISTER					
* (X) = REGISTER ADDRESS ON STACK					
* (DIGADD) INITIALIZED TO DIGIT 6					
* USES: ALL,TO					
FD4F	8D 3B	REGX	BSR	OUTSTJ	PRINT 'REGX'
FD51	30 95		FCC	LTR1,LTRN+#80	
FD53	20 16		BRA	REGX1	
FD55	8D 35	REGA	BSR	OUTSTJ	PRINT 'ACCA'
FD57	77 0D 0D		FCC	HEXA,LTRC,LTRC,LTRA+#80	
FD5B	20 10		BRA	REGA1	
FD5D	8D 2D	REGB	BSR	OUTSTJ	PRINT 'ACCB'
FD5F	77 0D 0D		FCC	HEXA,LTRC,LTRC,LTRB+#80	
FD63	20 09		BRA	REGB1	
FD65	8D 25	REGP	BSR	OUTSTJ	PRINT 'PC'
FD67	67 8D		FCC	LTRP,LTRC+#80	
FD69	4C		INC A		(A) = OFFSET INTO STACK
FD6A	4C		INC A		(B) = #BYTES THIS REGISTER
FD6B	5C	REGX1	INC B		
FD6C	4C		INC A		
FD6D	4C	REGA1	INC A		
FD6E	5C	REGB1	INC B		
FD6F	8B 02		ADD A	#2	



```

FD71  DE F2      LDX  USERS
FD73  08          REG1 INX
FD74  4A          DEC A      POINT X TO REGISTER
FD75  26 FC      BNE  REG1
FD77  8D 02      BSR  DISPLAY
FD79  4C          INC A
FD7A  39          RTS

```

```

**      DISPLAY - DISPLAY INDEXED BYTES
*
*      ENTRY: (X) = ADDRESS OF BYTES TO OUTPUT
*              (B) = NUMBER OF BYTES TO DISPLAY
*      EXIT: X,B UNCHANGED
*              (DIGADD) UPDATED
*      USES: ALL, TO

```

```

FD7B  37          DISPLAY FSH B
FD7C  A6 00      DIS1 LDA A  0,X      GET BYTE
FD7E  BD FE 20   JSR  OUTBYT      DISPLAY BYTE
FD81  08          INX
FD82  5A          DEC B
FD83  26 F7      BNE  DIS1
FD85  33          PUL B
FD86  17          TBA          DUPLICATE BYTE COUNT
FD87  09          DIS2 DEX          RESTORE X
FD88  4A          DEC A
FD89  26 FC      BNE  DIS2
FD8B  39          RTS

```

```

*      CLEAR B AND JUMP TO OUTSTR

```

```

FD8C  5F          OUTSTJ CLR B
FD8D  CE C1 6F   OUTSTO LDX  #DIGADD
FD90  7E FE 50   JMP  OUTST1

```

```

**      CONDX - DISPLAY CONDITION CODES
*
*      ENTRY: DIGADD INITIALIZED
*      EXIT:  (B) = 0
*      USES: ALL, TO

```

```

FD93  BD FC BC   CONDX JSR  REDIS      RESET DISPLAYS
FD96  DE F2      LDX  USERS
FD98  C6 20      LDA B  #20
FD9A  4F          CONDO CLR A
FD9B  E5 01      BIT B  1,X          MASK DESIRED BIT
FD9D  27 01      BEQ  COND1          IS A ZERO
FD9F  4C          INC A              IS A ONE
FDA0  BD FE 28   COND1 JSR  OUTHEX
FDA3  56          ROR B
FDA4  26 F4      BNE  CONDO          MORE TO GO
FDA6  4C          INC A
FDA7  39          RTS

```

```

**      STKPTR - OUTPUT USER STACK POINTER
*
*      ENTRY: (DIGADD) INITIALIZED
*      EXIT:  (B) = 0
*      USES: ALL, TO

```

```

FDA8          REGS EQU  *
FDA8  8D E2     STKPTR BSR  OUTSTJ
FDDA  5B E7     FCC  LTRS,LTRP+#80
FDAC  D6 F3     LDA B  USERS+1
FDAE  CE 07     ADD B  #7

```




```

FDB0 99 F2      ADC A  USERS      CLEAN UP FOR USER
FDB2 8D 6C      BSR      OUTBYT
FDB4 17          TRA
FDB5 5F          CLR B
FDB6 8D 68      BSR      OUTBYT
FDB8 86 01      LDA A  #1
FDBA 39          RTS

```

** ENCODE - SCAN AND ENCODE KEYBOARD

*

* ENTRY: NONE

* EXIT: (A) = HEX VALUE OF KEY PRESSED

* 'C' SET FOR VALID CONDITION

* USES: A,C,T0

```

FDBB 37          ENCODE PSH B
FDBC F6 C0 03    LDA B  COL1      GET KEYBOARD DATA
FDBF B6 C0 06    LDA A  COL3
FDC2 48          ASL A      LEFT JUSTIFY DATA
FDC3 48          ASL A
FDC4 48          ASL A
FDC5 59          ROL B
FDC6 48          ASL A
FDC7 59          ROL B
FDC8 48          ASL A
FDC9 59          ROL B
FDCA 37          PSH B
FDCB F6 C0 05    LDA B  COL2      GET LAST DATA
FDCE C4 1F      AND B  #1F      MASK ANY GARBAGE
FDD0 1B          ABA
FDD1 33          PUL B
FDD2 43          COM A
FDD3 53          COM B

```

* (BA) IS NOW KEYBOARD PATTERN

```

FDD4 DF EC      STX      T0
FDD6 CE FF A5    LDX      #HEXTAB-1    TABLE OF POSSIBLE OUTPUTS
FDD9 11          CBA      FIND ACTIVE ACCUMULATOR
FDDA 27 11      BEQ      ENC3    ILLEGAL OR NO KEY
FDDC 24 06      BCC      ENC1    A ACTIVE
FDD E 36      PSH A      B ACTIVE
FDDF 17          TRA      INTERCHANGE B,A
FDE0 33          PUL B
FDE1 CE FF AD    LDX      #HEXTAB+7
FDE4 5D          ENC1    TST B      B SHOULD BE ZERO
FDE5 26 06      BNE      ENC3    ILLEGAL
FDE7 08          ENC2    INX      SCAN FOR ACTIVE BIT
FDE8 48          ASL A
FDE9 22 FC      RHI      ENC2    NOT ACTIVE BIT
FDEB 27 01      BEQ      ENC4    LEGAL CHARACTER
FDED 0C          ENC3    CLC      ILLEGAL RETURNS 'C' CLEAR
FDEE A6 00      ENC4    LDA A  0,X    GET HEX FROM TABLE
FDF0 DE EC      LDX      T0
FDF2 33          PUL B      CLEAN UP
FDF3 39          RTS      AND RETURN

```

** INCH - INPUT CHARACTER FROM KEYBOARD

*

* 'INCH' WAITS FOR A TRANSITION BETWEEN ILLEGAL AND
 * LEGAL KEYBOARD CONDITIONS, AND RETURNS HEX VALUE
 * OF KEY DEPRESSED

* ENTRY: NONE

* EXIT: (A) = HEX VALUE

* USES: A,C,T0

```

FDF4 37          INCH  PSH B
FDF5 C6 7F      INC1  LDA B  #TIME    VIOLATION COUNT

```



```

FDF7 8D C2      INC2   BSR      ENCODE      WAIT FOR ILLEGAL INTERVAL
FDF9 25 FA      BCS      INC1      STILL LEGAL
FDFB 5A         DEC B      BNE      INC2      NOT A FELONY
FDFF 26 F9

*              NOW WE'RE SURE WE HAVE AN ILLEGAL CONDITION AND
*              NOT JUST A RELEASE CONTANT BOUNCE

FDFF 06 7F      INC3   LDA B      #TIME      TIME UNTIL PAROLE
FE00 8D B9      INC4   BSR      ENCODE
FE02 24 FA      BCC      INC3      BAD BEHAVIOR
FE04 5A         DEC B      BNE      INC4      BACK IN THE SLANNER
FE05 26 F9      PUL B
FE07 33         RTS
FE0B 39

**            INH - INPUT HEX BYTE AND DISPLAY ON LEDS
*
*            ENTRY: NONE
*            EXIT:  (A) = BYTE VALUE
*                   (DIGADD) UPDATED
*            USES:  A,T0,C

FE09 8D E9      INH     BSR      INCH      GET FIRST HALF
FE0B 8D 1B      BSR      OUTHEX     ECHO TO DISPLAYS
FE0D 48         ASL A
FE0E 48         ASL A
FE0F 48         ASL A
FE10 48         ASL A
FE11 37         PSH B
FE12 16         TAB
FE13 8D DF      BSR      INCH      GET NEXT HALF
FE15 8D 11      BSR      OUTHEX     ECHO
FE17 1B         ABA
FE18 33         PUL B
FE19 36         PSH A
FE1A 8D 9F      INH1    BSR      ENCODE      WAIT FOR KEY RELEASE
FE1C 25 FC      BCS      INH1
FE1E 32         PUL A      RESTORE LEGAL ENTRY
FE1F 39         RTS

**            OUTBYT - OUTPUT TWO HEX DIGITS
*
*            ENTRY: (A) = BYTE VALUE TO OUTPUT
*            EXIT:  (DIGADD) UPDATED
*            USES:  C,T0

FE20 36         PSH A
FE21 44         LSR A
FE22 44         LSR A
FE23 44         LSR A
FE24 44         LSR A
FE25 8D 01      BSR      OUTHEX     OUTPUT N.S. FOUR BITS
FE27 32         PUL A

**            OUTHEX - OUTPUT HEX DIGIT
*
*            ENTRY: (A) = HEX VALUE
*            EXIT:  (DIGADD) UPDATED
*            USES:  C,T0

FE28 36         PSH A
FE29 84 0F      AND A      #F      MASK GARBAGE
FE2B DF EC      STX      T0
FE2D CE FF 95   LDX      #DISTAB-1  DISPLAY CODE TABLE
FE30 0B         INX
FE31 4A         DEC A
FE32 2A FC      BPL      OUTH1
FE34 A6 00      LDA A      0,X      DISPLAY CODE FOR HEX
FE36 8D 04      BSR      OUTO      ALTERNATE ENTRY FOR 'OUTCH'

```



FE38 32
FE39 39

PUL A
RTS

```

**      OUTCH - OUTPUT CHARACTER TO DISPLAY
*
*      ENTRY: (A) = SEGMENT CODE
*              (DIGADD) = ADDRESS OF DIGIT TO OUTPUT
*      EXIT:  (DIGADD) UPDATED
*      USES: C,TO

```

```

FE3A DF EC      OUTCH STX      TO
FE3C DE F0      OUT0  LDX      DIGADD      **ALTERNATE ENTRY** FROM 'OUTHEX'
FE3E 37         PSH B
FE3F 49         ROL A
FE40 49         ROL A      PRE-ROTATE A
FE41 C6 10      LDA B      #10      TO GET TO NEXT DIGIT
FE43 49         ROL A      HERE WE MAKE TWO PASSES AT
FE44 A7 00      STA A      0,X      LIGHTING DIGITS--
FE46 09         DEX              KING'S X ON FIRST PASS!!
FE47 5A         DEC B
FE48 26 F9      BNE      OUT1
FE4A DF F0      STX      DIGADD      UPDATE 'DIGADD'
FE4C DE EC      LDX      TO      RESTORES X
FE4E 33         PUL B
FE4F 39         RTS

```

```

**      OUTSTR--OUTPUT INBEDDED CHARACTER STRING
*      CALLING CONVENTION:
*      JSR      OUTSTR
*      FIRST CHARACTER
*              *
*              *
*      LAST CHARACTER (HAS D.P. LIT)
*      NEXT INSTRUCTION
*
*      ENTRY:  NONE
*      EXIT:   TO 'NEXT INSTRUCTION'
*              (A) = 0
*      USES:   A,X,TO

```

```

FE50 DF F0      OUTST1 STX      DIGADD      **ALTERNATE ENTRY** SETS UP DIGI
FE52 30         OUTSTR TSX
FE53 EE 00      LDX      0,X      POINT 'X' AT STRING
FE55 31         INS
FE56 31         INS
FE57 A6 00      LDA A      0,X      GET CHARACTER
FE59 8D DF      BSR      OUTCH      OUTPUT IT TO DISPLAYS
FE5B 0B         INX
FE5C 4D         TST A      LAST CHARACTER IS NEGATIVE
FE5D 2A F8      BPL      OUTST3
FE5F 4F         CLR A
FE60 6E 00      JMP      0,X      RETURN TO 'NEXT INST.'

```

```

**      STEP - STEP USER CODE
*
*      ENTRY:  NONE
*      EXIT:   (B) = 1
*              (X) = USER P.C.
*              (A) = 0
*      USES:   ALL,TO,T1

```

```

FE62 8D 07      STEP  BSR      SSTEP      STEP USER CODE
FE64 DE F2      LDX      USERS      DISPLAY INSTRUCTION
FE66 EE 06      LDX      6,X
FE68 7E FC F9   JMP      MEM

```

```

**      SSTEP - PERFORM SINGLE STEP.
*

```



FE6B	9F EE	SSTEP	STS	TEMP	WE'LL USE THIS WHEN WE RETURN
FE6D	DE F2		LDX	USERS	
FE6F	A6 07		LDA A	7,X	PUSHING USER PC ONTO MONITOR
FE71	36		PSH A		STACK
FE72	A6 06		LDA A	6,X	
FE74	36		PSH A		
FE75	EE 06		LDX	6,X	NOW GET USER PC INTO X
FE77	86 3F		LDA A	##3F	SWI'S ARE NORMAL EXIT FROM
FE79	36		PSH A		SCRATCHPAD EXECUTION
FE7A	36		PSH A		
FE7B	A6 02		LDA A	2,X	NOW WE ARE COPYING THREE BYTES
FE7D	36		PSH A		OF INSTRUCTION
FE7E	A6 01		LDA A	1,X	
FE80	36		PSH A		
FE81	A6 00		LDA A	0,X	THIS IS THE OP CODE SO
FE83	36	BYTCNT	PSH A		SCRUTINIZE CAREFULLY
FE84	16		TAB		
FE85	CE FF 75		LDX	#OFTAB-1	
FE88	08	BYT1	INX		
FE89	C0 08		SUB B	#8	
FE8B	24 FB		BCC	BYT1	
FE8D	A6 00		LDA A	0,X	
FE8F	46	BYT2	ROR A		
FE90	5C		INC B		
FE91	26 FC		BNE	BYT2	
FE93	32		PUL A		
FE94	36		PSH A		
FE95	25 1E		BCS	BYT7	
FE97	81 30		CMP A	##30	CHECK FOR BRANCH
FE99	24 04		BCC	BYT3	
FE9B	81 20		CMP A	##20	
FE9D	24 14		BCC	BYT5	IT IS A BRANCH
FE9F	81 60	BYT3	CMP A	##60	
FEA1	25 11		BCS	BYT6	IT IS ONE BYTE
FEA3	81 8D		CMP A	##8D	
FEA5	27 0C		BEQ	BYT5	IT IS BSR
FEA7	84 BD		AND A	##BD	
FEA9	81 8C		CMP A	##8C	
FEAB	27 04		BEQ	BYT4	IS X OR SP IMMEDIATE
FEAD	84 30		AND A	##30	CHECK FOR THREE BYTES
FEAF	81 30		CMP A	##30	
FEB1	C2 FF	BYT4	SBC B	##FF	
FEB3	5C	BYT5	INC B		
FEB4	5C	BYT6	INC B		
FEB5	27 70	BYT7	BEQ	BSTRD	
FEB7	30		TSX		
FEB8	25 02		BCS	STEP1	
FEBA	E7 01		STA B	1,X	BRANCH OFFSET TO 2
FEBB	86 01	STEP1	LDA A	#1	
FEBE	C1 02		CMP B	#2	
FEC0	2E 06		BGT	STEP3	
FEC2	27 02		BEQ	STEP2	TWO BYTES
FEC4	A7 01		STA A	1,X	FOR ONE BYTES
FEC6	A7 02	STEP2	STA A	2,X	NOT FOR THREE BYTES
FEC8	4F	STEP3	CLR A		NOW ADD BYTE COUNT TO PC
FEC9	EB 06		ADD B	6,X	
FECB	A9 05		ADC A	5,X	
FECD	A7 05		STA A	5,X	
FECF	E7 06		STA B	6,X	
* DOES THE INSTRUCTION INVOLVE THE PC? IF SO THEN IT					
* MUST BE INTERPRETED					
FED1	DE F2	SRCHOP	LDX	USERS	
FED3	A7 06		STA A	6,X	
FED5	E7 07		STA B	7,X	UPDATE PC ON USER STACK
FED7	C6 06		LDA B	#6	
FED9	32		PUL A		
FEDA	36		PSH A		



```

FEDB 84 CF      AND A  #$CF      IS THIS A SUBROUTINE CALL?
FEDD 81 8D      CMP A  #$8D
FEDF 32         PUL A
FEE0 27 48      BEQ     BSRH
FEE2 81 6E      CMP A  #$6E
FEE4 27 5B      BEQ     JPXH      IT IS INDEXED JUMP
FEE6 81 7E      CMP A  #$7E
FEE8 27 5E      BEQ     JMPH      IT IS EXTENDED JUMP
FEEA 81 39      CMP A  #$39
FEEC 27 62      BEQ     RTSH      IT IS RTS
FEEE 81 3B      CMP A  #$3B
FEF0 27 6C      BEQ     RTIH      IT IS RTI
FEF2 81 3F      CMP A  #$3F
FEF4 27 6E      BEQ     SWIH      IT IS SWI
FEF6 AF 06      STS     6,X      AIM USER PC AT SCRATCH AREA
FEF8 36         PSH A      REPLACE OPCODE
FEF9 CE FF 05   LDX     #SSRET

**          SWIVE1 - SET UP BREAKPOINT RETURN AND JUMP TO USER CODE
*
*          ENTRY:  (X) = SWI VECTOR
*          EXIT:   TO USER PROGRAM

FEFC 86 7E      SWIVE1 LDA A  #$7E      JUMP OF CODE
FEFE 97 F4      STA A  SYSSWI
FF00 DF F5      STX     SYSSWI+1
FF02 9E F2      LDS     USERS
FF04 3B         RTI

*          THE FOLLOWING CODE IS EXECUTED AFTER A SINGLE STEP
*          OF AN OUT-OF-PLACE INSTRUCTION.  NOW CHECK TO SEE
*          IF BRANCH OCCURRED, MODIFY THE USER PC ACCORDINGLY

FF05 30         SSRET  TSX           GET SWI HIT LOCATION INTO X
FF06 EE 05      LDX     5,X
FF08 08         INX
FF09 4F         CLR A
FF0A 5F         CLR B
FF0B 9C EE      CPX     TEMP
FF0D 26 0C      BNE     BCHNTR

*          ADD THE BRANCH OFFSET TO THE USER PC

FF0F 09         DEX           X WILL NOW POINT AT USERPC
FF10 EE 00      LDX     0,X      SAVED VALUE OF PC INTO X
FF12 09         DEX           PREPARE TO FETCH BRANCH OFFSET
FF13 E6 00      LDA B  0,X
FF15 2A 01      BPL     PLUS
FF17 43         COM A           A IS SIGN EXTENSION OF B
FF18 30         PLUS  TSX      LO COST WAY TO POINT TO USERPC
FF19 EE 05      LDX     5,X
FF1B EB 01      BCHNTR ADD B  1,X      ADD BRANCH OFFSET OR ZERO TO PC
FF1D A9 00      ADC A  0,X
FF1F 30         TSX           PLACE NEW USERPC ONTO STACK
FF20 A7 05      STA A  5,X
FF22 E7 06      STA B  6,X
FF24 09         DEX           NOW X AND SP ARE EQUAL
FF25 DF F2      STOX  STX     USERS
FF27 9E EE      BSTRD LDS     TEMP
FF29 39         RTS           RETURN TO CALLING ROUTINE

**          SPECIAL HANDLERS

```



```

**      JSR HANDLER

FF30  80 3F      JSRH      SUB A  -#3F      JSR'S TO JUMPS
FF32  36         PSH A      CORRECTED OPCODE ONTO STACK
FF33  09         DEX
FF34  09         DEX
FF35  DF F2      STX        USERS
FF37  A6 03      JSRH1     LDA A  3,X
FF39  A7 01      STA A  1,X      MOVE USER REGISTERS
FF3B  08         INX
FF3C  5A         DEC B
FF3D  2A F8      BPL        JSRH1
FF3F  20 90      BRA        SRCHOP      NOW EXECUTE JUMP INSTRUCT

```

```

**      JFXH - INDEXED JUMP HANDLER.

FF41  33         JFXH      PUL B      GET OFFSET
FF42  4F         CLR A
FF43  EB 05      ADD B  5,X
FF45  A9 04      ADC A  4,X
FF47  8C         FCB        $8C      CFXH: ONE BYTE BRA NEWPC

```

```

**      JMP HANDLER

FF48  32         JMPH      PUL A
FF49  33         PUL B
FF4A  A7 06      NEWPC     STA A  6,X
FF4C  E7 07      STA B  7,X
FF4E  20 D5      BRA        STOX      RETURN TO CALLER

```

```

**      RTS HANDLER

FF50  08         RTSH      INX
FF51  08         INX
FF52  DF F2      STX        USERS      NET PULL OF TWO BYTES
FF54  A6 03      RTS1     LDA A  3,X      MOVE FIVE BYTES
FF56  A7 05      STA A  5,X
FF58  09         DEX
FF59  5A         DEC B
FF5A  2E F8      BGT        RTS1
FF5C  20 C9      BRA        BSTRD

```

```

**      RTI HANDLER

FF5E  08         RTIH      INX
FF5F  5A         DEC B
FF60  2A FC      BPL        RTIH
FF62  20 C1      BRA        STOX

```

```

**      SWI HANDLER

FF64  A6 07      SWIH      LDA A  7,X
FF66  A7 00      STA A  0,X
FF68  09         DEX
FF69  5A         DEC B
FF6A  2A F8      BPL        SWIH
FF6C  8A 10      ORA A  #$00010000      SET INTERRUPT MASK
FF6E  A7 01      STA A  1,X
FF70  C6 FA      LDA B  #-USWI/256*256+USWI      USWI LO ORDER
FF72  86 00      LDA A  #$USWI/256
FF74  20 D4      BRA        NEWPC      PATCH IN UIRG

```



** OPTAB - LEGAL OP-CODE LOOKUP TABLE

FF76	9C 00 3C	OPTAB	FDB	\$9C00,\$3CAF,\$4000,\$00AC,\$6412,\$6412,\$6410,\$6410
FF86	11 01 10		FDB	\$1101,\$1004,\$1000,\$1000,\$110D,\$100C,\$100C,\$100C

** HEX DISPLAY CODE TABLE

FF96	7E 30 6D	DISTAB	FCC	HEX0,HEX1,HEX2,HEX3,HEX4,HEX5,HEX6,HEX7
FF9E	7F 7B 77		FCC	HEX8,HEX9,HEXA,HEXB,HEXC,HEXD,HEXE,HEXF

** KEY VALUE TABLE

FFA6	07 0A 0D	HEXTAB	FCC	7,10,13,2,5,8,11,14
FFAE	03 06 09		FCC	3,6,9,12,15,0,1,4

** COMMAND HANDLER ENTRY POINT TABLE

FFB6	FC 45 FD	CMDTAB	FDB	ZERO,REGA,REGB,REGP,REGX,CONDX,REGS,RESUME,STEP
FFC8	FC 46 FD		FDB	BKSET,AUTO,BACK,REPLAC,DO,EXAM,FOWD

FFF8		ORG	\$FFF8
------	--	-----	--------

** INTERRUPT VECTORS.

FFF8	00 F7	FDB	UIRQ	USER IRQ HANDLER
FFFA	00 F4	FDB	SYSSWI	SYSTEM SWI HANDLER
FFFC	00 FD	FDB	UNMI	USER NMI HANDLER
FFFE	FC 00	FDB	RESET	
0000		END		

STATEMENTS =970

FREE BYTES =24298

NO ERRORS DETECTED



SYMBOLIC REFERENCE TABLE.

ADDR	FC7E	DIGADD	00F0	MEM	FCF9	REG1	FD73
AUTO	FD0A	DISTAB	FF96	MEM2	FD07	REPLAC	FD1B
AUT1	FD0C	DIS1	FD7C	MONSTK	00E3	REPL1	FD24
BACK	FD18	DIS2	FD87	NEWPC	FF4A	RESET	FC00
BADDR	FC06	DO	FC8C	OPTAB	FF76	RESE1	FC15
BCHNTK	FF1B	DOPMT	FC6F	OUTBYT	FE2D	RESUME	FC96
BKPT	FC0E	DOPMI	FC75	OUTCH	FE3A	RES1	FC98
BKP1	FC07	DSPLAY	FD7B	OUTHEX	FE28	RES2	FCA3
BKP2	FCE1	ENCODE	FD8B	OUTH1	FE30	RES3	FCA8
BKP3	FCE6	ENC1	FDE4	OUTSTA	FC86	RTIH	FF5E
BKP4	FCF0	ENC2	FDE7	OUTSTJ	FD8C	RTSH	FF50
BKSET	FC46	ENC3	FDED	OUTSTR	FE52	RTS1	FF54
BKSE1	FC4D	ENC4	FDEE	OUTSTO	FD8D	SRCHOP	FED1
BKSE2	FC57	EXAM	FD13	OUTST1	FE50	SSRET	FF05
BKSE3	FC65	POWD	FD16	OUTST3	FE57	SSTEP	FE68
BKSP	FD43	HEXTAB	FFA6	OUTO	FE3C	STEP	FE62
BKSP1	FD46	IHB	FE09	OUT1	FE43	STEP1	FEBC
BSRH	FE2A	IHB1	FE1A	PLUS	FF18	STEP2	FE06
BSTRD	FF27	INCH	FD4F	PROMPT	FD25	STEP3	FE08
BYTCNT	FE83	INC1	FD55	PROM1	FD29	STKPTR	FDA8
BYT1	FE88	INC2	FD57	PROM2	FD33	STOX	FF25
BYT2	FE8F	INC3	FD5E	PROM3	FD3E	SWIH	FF64
BYT3	FE9F	INC4	FE00	REDIS	FC8C	SWIVE1	FEFC
BYT4	FE81	JMPH	FF48	REGA	FD55	SYSSWI	00F4
BYT5	FE83	JPXH	FF41	REGA1	FD6D	ZERO	FC45
BYT6	FE84	JSRH	FF30	REG8	FD5D		
BYT7	FE85	JSRH1	FE37	REG81	FD6E		
CMDTAB	FFB6	MAIN	FC19	REGP	FD65		
CONDX	FD93	MAIN1	FC21	REGS	FDA8		
CONDO	FD9A	MAIN2	FC31	REGX	FD4F		
COND1	FDA0	MAIN3	FC36	REGX1	FD6B		

ASSEMBLY CONSTANT TABLE

KEYBOARD LOCATIONS

COL1	EQU	\$0003	RIGHTMOST COLUMN
COL2	EQU	\$0005	
COL3	EQU	\$0006	LEFTMOST COLUMN

MISC. CONSTANTS

TIME	EQU	32	
NBR	EQU	4	NUMBER BREAKPOINTS

*

*

DISPLAY ADDRESSES

DG6ADD	EQU	\$C16F	LEFTMOST DIGIT
DG5ADD	EQU	\$C15F	
DG4ADD	EQU	\$C14F	
DG3ADD	EQU	\$C13F	
DG2ADD	EQU	\$C12F	
DG1ADD	EQU	\$C11F	RIGHTMOST DIGIT

*

*

DISPLAYED CHARACTER SEGMENT CODES

HEX0	EQU	\$7E	HEX8	EQU	\$7F	LTRA	EQU	\$7D	LTRH	EQU	\$37
HEX1	EQU	\$30	HEX9	EQU	\$7B	LTRB	EQU	\$1F	LTRD	EQU	\$3D
HEX2	EQU	\$6D	HEXA	EQU	\$77	LTRC	EQU	\$0D	LTRG	EQU	\$5E
HEX3	EQU	\$79	HEXB	EQU	\$1F	LTRF	EQU	\$47	LTR0	EQU	\$1D
HEX4	EQU	\$33	HEXC	EQU	\$4E	LTRN	EQU	\$15	LTRR	EQU	\$05
HEX5	EQU	\$5B	HEXD	EQU	\$3D	LTRI	EQU	\$30	LTRU	EQU	\$3E
HEX6	EQU	\$5F	HEXE	EQU	\$4F	LTRP	EQU	\$67	LTRY	EQU	\$3B
HEX7	EQU	\$70	HEXF	EQU	\$47	LTRL	EQU	\$0E	LTRS	EQU	\$5B
									DASH	EQU	\$08

RESERVED MEMORY BYTES IN RAM

00CB	JSRSTK	EQU	*-6	
00D1		RMB	19	
00E3	MONSTK	EQU	*-1	
00E4	BKTBL	RMB	2*NBR	
00EC	TO	RMB	2	TEMPORARY
00EE	TEMP	RMB	2	USFD BY SINGLE STEPPER
00F0	DIGADD	RMB	2	DISPLAY POINTER
00F2	USERS	RMB	2	USER STACK POINTER
00EF	T1	EQU	TEMP	
00F4	SYSSWI	RMB	3	SYSTEM SWI VECTOR
00F7	IRQ	RMB	3	USER IRQ VECTOR
00FA	USWI	RMB	3	USER SWI VECTOR
00FD	UNMI	RMB	3	USER NMI VECTOR



MEMORY

Memory Map

The memory is organized as shown below.

Monitor ROM	FFFF FC00
Not usable	C1FF C170
Displays	C16F * C110
Not usable	C10F C00F
Keyboard	C00E * C003
Not usable	C002 C000
Optional 256 bytes of user RAM	01FF 0100
59 bytes RAM (reserved for monitor)	00FF 00C5
197 bytes of user RAM	00C4 0000

Memory Decoding

	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
ROM IC12 FFXX FCxx	1	1	1	1	1	1	X	X	X	X	X	X	X	X	X	X
RAM (Optional) IC16, IC17 01XX	0	0	0	0	0	0	0	1	X	X	X	X	X	X	X	X
RAM IC14, IC15 00XX	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
KEYBOARD C0 - X	1	1	0	0	0	0	0	0	—	—	—	—	—	X	X	X
DISPLAYS C1XX	1	1	0	0	0	0	0	1	—	X	X	X	—	X	X	X

1 = LOGIC 1, 0 = LOGIC 0, — = DOES NOT CARE, X = FUNCTIONING ADDRESS

* Not fully decoded.

Keyboard And Display Functioning Addresses

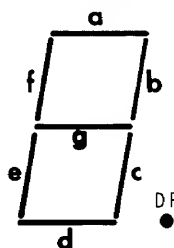
KEYBOARD

Keys	BINARY				HEX
	A ₃	A ₂	A ₁	A ₀	
3, 6, 9, C, F	—	0	1	1	3
2, 5, 8, B, E	—	1	0	1	5
0, 1, 4, 7, A, D	—	1	1	0	6

DISPLAY

LED	BINARY			HEX
	A ₆	A ₅	A ₄	
H	1	1	0	6
I	1	0	1	5
N	1	0	0	4
Z	0	1	1	3
V	0	1	0	2
C	0	0	1	1

LED SEGMENTS



LED SEGMENT	BINARY				HEX
	A ₃	A ₂	A ₁	A ₀	
a	—	1	1	0	E or 6
b	—	1	0	1	D or 5
c	—	1	0	0	C or 4
d	—	0	1	1	B or 3
e	—	0	1	0	A or 2
f	—	0	0	1	9 or 1
g	—	0	0	0	8 or 0
DP	—	1	1	1	F or 7

— = DOES NOT CARE



INSTRUCTION SET*

ACCUMULATOR AND MEMORY		ADDRESSING MODES															BOOLEAN/ARITHMETIC OPERATION (All register labels refer to contents)	COND. CODE REG.										
		IMMED			DIRECT			INDEX			EXTND			INHER				5	4	3	2	1	0					
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C					
OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#												
Add	ADDA	88	2	2	98	3	2	A8	5	2	B8	4	3				A + M → A		↑	•	↑	↑	↑	↑				
	ADDB	C8	2	2	D8	3	2	E8	5	2	F8	4	3				B + M → B		↑	•	↑	↑	↑	↑				
Add Acmltrs	ABA													1B	2	1	A + B → A		↑	•	↑	↑	↑	↑				
Add with Carry	ADCA	89	2	2	99	3	2	A9	5	2	B9	4	3				A + M + C → A		↑	•	↑	↑	↑	↑				
	ADCB	C9	2	2	D9	3	2	E9	5	2	F9	4	3				B + M + C → B		↑	•	↑	↑	↑	↑				
And	ANDA	84	2	2	94	3	2	A4	5	2	B4	4	3				A • M → A		•	•	↑	↑	R	•				
	ANDB	C4	2	2	D4	3	2	E4	5	2	F4	4	3				B • M → B		•	•	↑	↑	R	•				
Bit Test	BITA	85	2	2	95	3	2	A5	5	2	B5	4	3				A • M		•	•	↑	↑	R	•				
	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3				B • M		•	•	↑	↑	R	•				
Clear	CLR							6F	7	2	7F	6	3				00 → M		•	•	R	S	R	R				
	CLRA													4F	2	1	00 → A		•	•	R	S	R	R				
	CLRB													5F	2	1	00 → B		•	•	R	S	R	R				
Compare	CMPA	81	2	2	91	3	2	A1	5	2	B1	4	3				A - M		•	•	↑	↑	↑	↑				
	CMPB	C1	2	2	D1	3	2	E1	5	2	F1	4	3				B - M		•	•	↑	↑	↑	↑				
Compare Acmltrs	CBA													11	2	1	A - B		•	•	↑	↑	↑	↑				
Complement, 1's	COM							63	7	2	73	6	3				M → M		•	•	↑	↑	R	S				
	COMA													43	2	1	A → A		•	•	↑	↑	R	S				
	COMB													53	2	1	B → B		•	•	↑	↑	R	S				
Complement, 2's (Negate)	NEG							60	7	2	70	6	3				00 → M + M		•	•	↑	↑	①	②				
	NEGA													40	2	1	00 → A + A		•	•	↑	↑	①	②				
	NEGB													50	2	1	00 → B + B		•	•	↑	↑	①	②				
Decimal Adjust, A	DAA													19	2	1	Converts Binary Add of BCD Characters into BCD Format		•	•	↑	↑	↑	③				
Decrement	DEC							6A	7	2	7A	6	3				M - 1 → M		•	•	↑	↑	④	•				
	DECA													4A	2	1	A - 1 → A		•	•	↑	↑	④	•				
	DECB													5A	2	1	B - 1 → B		•	•	↑	↑	④	•				
Exclusive OR	EDRA	8B	2	2	9B	3	2	AB	5	2	BB	4	3				A ⊕ M → A		•	•	↑	↑	R	•				
	EDRB	CB	2	2	DB	3	2	EB	5	2	FB	4	3				B ⊕ M → B		•	•	↑	↑	R	•				
Increment	INC							6C	7	2	7C	6	3				M + 1 → M		•	•	↑	↑	⑤	•				
	INCA													4C	2	1	A + 1 → A		•	•	↑	↑	⑤	•				
	INCB													5C	2	1	B + 1 → B		•	•	↑	↑	⑤	•				
Load Acmltr	LDAA	86	2	2	96	3	2	A6	5	2	B6	4	3				M → A		•	•	↑	↑	R	•				
	LDAB	C6	2	2	D6	3	2	E6	5	2	F6	4	3				M → B		•	•	↑	↑	R	•				
Or, Inclusive	ORAA	8A	2	2	9A	3	2	AA	5	2	BA	4	3				A + M → A		•	•	↑	↑	R	•				
	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3				B + M → B		•	•	↑	↑	R	•				
Push Data	PSHA													36	4	1	A ← M _{SP} , SP - 1 → SP		•	•	•	•	•	•				
	PSHB													37	4	1	B ← M _{SP} , SP - 1 → SP		•	•	•	•	•	•				
Pull Data	PULA													32	4	1	SP + 1 → SP, M _{SP} → A		•	•	•	•	•	•				
	PULB													33	4	1	SP + 1 → SP, M _{SP} → B		•	•	•	•	•	•				
Rotate Left	ROL							69	7	2	79	6	3				M		•	•	↑	↑	⑥	↑				
	ROLA													49	2	1	A		•	•	↑	↑	⑥	↑				
	ROLB													59	2	1	B		•	•	↑	↑	⑥	↑				
Rotate Right	ROR							66	7	2	76	6	3				M		•	•	↑	↑	⑥	↑				
	RORA													46	2	1	A		•	•	↑	↑	⑥	↑				
	RORB													56	2	1	B		•	•	↑	↑	⑥	↑				
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M		•	•	↑	↑	⑥	↑				
	ASLA													48	2	1	A		•	•	↑	↑	⑥	↑				
	ASLB													58	2	1	B		•	•	↑	↑	⑥	↑				
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M		•	•	↑	↑	⑥	↑				
	ASRA													47	2	1	A		•	•	↑	↑	⑥	↑				
	ASRB													57	2	1	B		•	•	↑	↑	⑥	↑				
Shift Right, Logic	LSR							64	7	2	74	6	3				M		•	•	R	↑	⑥	↑				
	LSRA													44	2	1	A		•	•	R	↑	⑥	↑				
	LSRB													54	2	1	B		•	•	R	↑	⑥	↑				
Store Acmltr	STAA				97	4	2	A7	6	2	B7	5	3				A → M		•	•	↑	↑	R	•				
	STAB				D7	4	2	E7	6	2	F7	5	3				B → M		•	•	↑	↑	R	•				
Subtract	SUBA	80	2	2	90	3	2	A0	5	2	B0	4	3				A - M → A		•	•	↑	↑	↑	↑				
	SUBB	C0	2	2	D0	3	2	E0	5	2	F0	4	3				B - M → B		•	•	↑	↑	↑	↑				
Subtract Acmltrs	SBA													10	2	1	A - B → A		•	•	↑	↑	↑	↑				
Subtr. with Carry	SBCA	82	2	2	92	3	2	A2	5	2	B2	4	3				A - M - C → A		•	•	↑	↑	↑	↑				
	SBCB	C2	2	2	D2	3	2	E2	5	2	F2	4	3				B - M - C → B		•	•	↑	↑	↑	↑				
Transfer Acmltrs	TAB													16	2	1	A → B		•	•	↑	↑	R	•				
	TBA													17	2	1	B → A		•	•	↑	↑	R	•				
Test, Zero or Minus	TST							6D	7	2	7D	6	3				M - 00		•	•	↑	↑	R	R				
	TSTA													4D	2	1	A - 00		•	•	↑	↑	R	R				
	TSTB													5D	2	1	B - 00		•	•	↑	↑	R	R				

*Copied with permission of Motorola.

INDEX REGISTER AND STACK

POINTER OPERATIONS

INDEX REGISTER AND STACK		IMMED			DIRECT			INDEX			EXTND			INHER				5	4	3	2	1	0
POINTER OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#	BOOLEAN/ARITHMETIC OPERATION	H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3				(X _H /X _L) - (M/M + 1)	•	•	•	•	•	•
Decrement Index Reg	DEX													09	4	1	X - 1 → X	•	•	•	•	•	•
Decrement Stack Ptr	DES													34	4	1	SP - 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	4	1	X + 1 → X	•	•	•	•	•	•
Increment Stack Ptr	INS													31	4	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LOX	CE	3	3	DE	4	2	EE	6	2	FE	5	3				M → X _H , (M + 1) → X _L	•	•	•	•	•	•
Load Stack Ptr	LOS	BE	3	3	9E	4	2	AE	6	2	BE	5	3				M → SP _H , (M + 1) → SP _L	•	•	•	•	•	•
Store Index Reg	STX				OF	5	2	EF	7	2	FF	6	3				X _H → M, X _L → (M + 1)	•	•	•	•	•	•
Store Stack Ptr	STS				9F	5	2	AF	7	2	BF	6	3				SP _H → M, SP _L → (M + 1)	•	•	•	•	•	•
Indx Reg → Stack Ptr	TXS													35	4	1	X - 1 → SP	•	•	•	•	•	•
Stack Ptr → Indx Reg	TSX													30	4	1	SP + 1 → X	•	•	•	•	•	•

JUMP AND BRANCH

OPERATIONS

JUMP AND BRANCH		RELATIVE			INDEX			EXTND			INNER			BRANCH TEST							5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#								H	I	N	Z	V	C																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch Always	BRA	20		4	2									None							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If Carry Clear	BCC	24		4	2									C = 0							•	•	•	•	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If Carry Set	BCS	25		4	2									C = 1							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If = Zero	BED	27		4	2									Z = 1							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If > Zero	BGE	2C		4	2									N ÷ V = 0							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If < Zero	BGT	2E		4	2									Z + (N ÷ V) = 0							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If Higher	BHI	22		4	2									C + Z = 0							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If < Zero	BLE	2F		4	2									Z + (N ÷ V) = 1							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If Lower Or Same	BLS	23		4	2									C + Z = 1							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If < Zero	BLT	2D		4	2									N ÷ V = 1							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If Minus	BMI	28		4	2									N = 1							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If Not Equal Zero	BNE	26		4	2									Z = 0							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If Overflow Clear	BVC	28		4	2									V = 0							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If Overflow Set	BVS	29		4	2									V = 1							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch If Plus	BPL	2A		4	2									N = 0							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Branch To Subroutine	BSR	8D		8	2																•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Jump	JMP					6E		4		2	7E		3	3								•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
Jump To Subroutine	JSR					AD		8		2	BD		9	3								•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
No Operation	NOP														01		2		1	Advances Prog. Cntr. Only							•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
Return From Interrupt	RTI														3B		10		1								•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
Return From Subroutine	RTS														39		5		1								•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
Software Interrupt	SWI														3F		12		1								•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
Wait for Interrupt	WAI														3E		9		1								•	•	•	•	•	•																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											

CONDITIONS CODE REGISTER

MNEMONIC	INHER			BOOLEAN OPERATION	CONDITION CODE REGISTER NOTES						
	OP	~	#		H	I	N	Z	V	C	
Clear Carry	CLC	0C	2	1	0 - C						①
Clear Interrupt Mask	CLI	0E	2	1	0 - I						②
Clear Overflow	CLV	0A	2	1	0 - V						③
Set Carry	SEC	0D	2	1	1 - C						④
Set Interrupt Mask	SEI	0F	2	1	1 - I						⑤
Set Overflow	SEV	0B	2	1	1 - V						⑥
Accmlr A - CCR	TAP	06	2	1	A - CCR						⑦
CCR - Accmlr A	TPA	07	2	1	CCR - A						⑧

LEGEND

OP Operation Code (Hexadecimal)

~ Number of MPU Cycles.

Number of Program Bytes.

+ Arithmetic Plus.

- Arithmetic Minus.

• Boolean AND.

M_{sp} Contents of memory location pointed to be Stack Pointer.

+ Boolean Inclusive OR.

• Boolean Exclusive OR.

M Complement of M.

+ Transfer Into.

0 Bit = Zero.

00 Byte = Zero.

H Half carry from bit 3

I Interrupt mask

N Negative (sign bit)

Z Zero (byte)

V Overflow, 2's complement

C Carry from bit 7

R Reset Always

S Set Always

1 Test and set if true, cleared otherwise

• Not Affected

CCR Condition Code Register

LS Least Significant

MS Most Significant

CONDITION CODE REGISTER NOTES:

(Bit set if test is true and cleared otherwise)

① (Bit V) Test Result = 10000000?

② (Bit C) Test Result = 00000000?

③ (Bit C) Test Decimal value of most significant BCD Character greater than nine? (Not cleared if previously set)

④ (Bit V) Test Operand = 10000000 prior to execution?

⑤ (Bit V) Test Operand = 01111111 prior to execution?

⑥ (Bit V) Test Set equal to result of N ← C after shift has occurred

⑦ (Bit N) Test Sign bit of most significant (MS) byte of result = 1?

⑧ (Bit V) Test 2's complement overflow from subtraction of LS bytes?

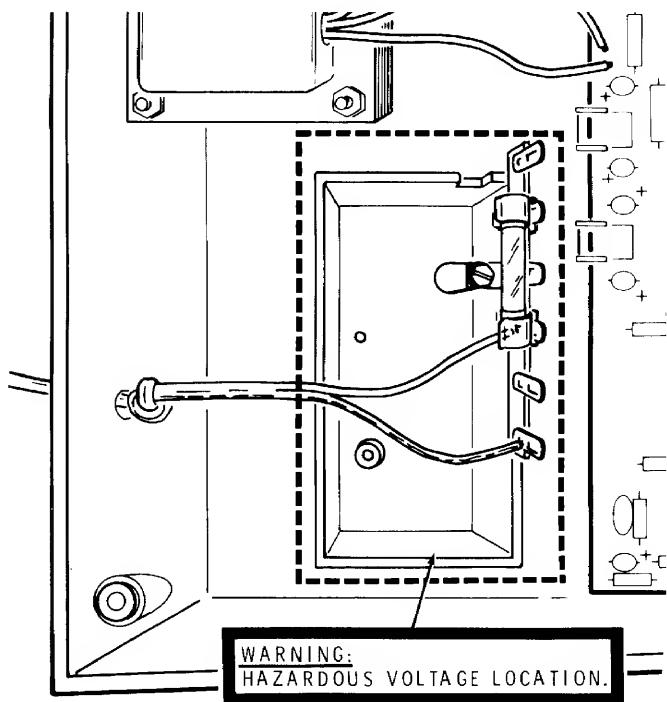
⑨ (Bit N) Test Result less than zero? (Bit 15 = 1)

⑩ (All) Load Condition Code Register from Stack (See Special Operations)

⑪ (Bit I) Set when interrupt occurs. If previously set, a Non Maskable Interrupt is required to exit the wait state.

⑫ (ALL) Set according to the contents of Accumulator A

IN CASE OF DIFFICULTY



PICTORIAL 9-1

WARNING: Dangerous AC voltage is present inside the cabinet (where the fuse is located) when the line cord is plugged in. See Pictorial 9-1.

This section of the Manual is divided into three parts: "Visual Checks," "Troubleshooting Chart," and "Detailed Troubleshooting." Use the "Visual Checks" first to find a difficulty that shows up right after your kit is assembled. You can also use the other two sections right after your kit is assembled, or at some future time — if your Trainer ever stops working.

If the trouble is still not located after you complete the "Visual Checks," check voltage readings against those shown in the "Schematic." NOTE: All voltage readings were taken with a high impedance voltmeter (10 M Ω or greater).

In the extreme case where you are unable to resolve a difficulty, refer to the "Customer Service" information inside the rear cover of your Manual. Your Warranty is located inside the front cover.

VISUAL TESTS

1. Recheck the wiring. Trace each lead in colored pencil on the Pictorial as it is checked. It is frequently helpful to have a friend check your work. Someone who is not familiar with the unit may notice something you consistently overlook.
2. About 90 percent of the kits that are returned to the Heath Company for repair do not function properly due to poor connections and soldering. Therefore, you can eliminate many troubles by reheating all connections to make sure they are soldered.
3. Check to be sure that all the integrated circuits are in their proper location and that each IC pin is properly installed in its connector, and not bent or under the IC.
4. Check the values of the parts. Be sure in each step that the proper part has been wired into the circuit, as shown in the Pictorial diagrams. It would be easy, for example, to install a 470 Ω (yellow-violet-brown) resistor where a 4700 Ω resistor (yellow-violet-red) resistor should have been installed.
5. Check for bits of solder, wire ends, or other foreign matter which may be lodged in the wiring.
6. A review of the "Theory of Operation" may also help you determine the trouble.

Precautions

1. Be cautious when you test IC's. Although they have almost unlimited life when used properly, they are much more vulnerable to damage from excessive voltage or current than some other components.
2. Be sure you do not short any terminals to ground when making voltage measurements. If the probe slips, for example, and shorts out a bias or supply point, it is very likely to damage one or more IC or diode.
3. Do not remove an IC while the line cord is plugged in.

Substitution

Corresponding display components can be interchanged; IC's 23 through 28 can be interchanged, for example. If one display unit shows a digit incorrectly, interchange it with one of the other units to determine if the display or the circuit is faulty. If the circuit is faulty and there are no solder bridges on the associated foil, interchange the decoder/driver IC with one of the others. This troubleshooting method can also be used with other problems.



TROUBLESHOOTING CHARTS

NOTES:

1. The following chart lists parts to check. These parts indicate areas of the circuits where problems could exist. Check the circuitry and look for an assembly error or **solder bridge**. Parts are rugged and reliable. Consider a part to be bad last.
2. If you make a repair, make sure you eliminate the cause as well as the effect of the trouble. If,

for example, you find a damaged part, be sure you find out what damaged the part. If the cause is not eliminated, the replacement part may also become damaged when you put the unit back into operation.

3. In several areas of the circuit boards, the foil patterns are quite narrow. When you unsolder a part to check or replace it, avoid excessive heat while you remove the part. A suction-type desoldering tool makes part removal easier.

POWER SUPPLIES

DIFFICULTY	POSSIBLE CAUSE
No +5 V, +12 V, and -12 V supplies. LED1 not lit.	<ol style="list-style-type: none"> 1. Fuse F1. 2. Transformer T1 primary wiring. 3. Line cord wiring.
No +5 V supply, in Standby or On position. LED1 not lit.	<ol style="list-style-type: none"> 1. Transformer T1 secondary wiring (green and green-yellow leads). 2*. Regulator IC31. 3. Short circuit on 5 V line.
No +5 V supply in On position.	<ol style="list-style-type: none"> 1. Switch SW1. 2. Short circuit on main 5 V line.
No +12 V supply in On position.	1.* Regulator IC29.
No -12 V supply in On position.	1.* Regulator IC30.
Have +5 V in Standby position. No +5 V in On position.	1. Shorted main +5 V line.
Have +5 V in Standby position. No +5 V to LED's in On position.	<ol style="list-style-type: none"> 1. Open main +5 V line. 2. Switch SW1.

*The voltage regulator IC's have built-in short circuit protection. Therefore, the lack of voltage at an output connector may indicate a short or open circuit on the circuit board or in the wiring.

Troubleshooting Chart (cont'd.)

7-SEGMENT LED's

DIFFICULTY	POSSIBLE CAUSE
No LED's light when "Segment Test" is shorted.	1. +5 V not supplied to LED's.
All seven segments of one LED do not light when "Segment test" is shorted.	1. +5 V not supplied to this LED.
One segment of an LED does not light when "Segment Test" is shorted.	1. LED segment. 2. Decoder driver. 3. LED not properly installed.
All segments of all LED's lit.	1. Clear line of decoder driver IC's shorted. 2. Jumper wire at terminal blocks A11/A12.
All LED's light dimly when "Segment Test" is shorted.	1. +5 V not supplied to IC's 23 through 28.
All LED's light when "Segment Test" is shorted, but one LED is dim.	1. +5 V not supplied to associated decoder driver IC. 2. Defective decoder driver IC.
All LED's light, except one segment, when "Segment Test" is shorted.	1. Associated LED. 2. Associated decoder driver IC.
One LED stays lit.	1. Associated LED. 2. Associated decoder driver IC.
LED's light dimly when experiments are connected and the Power switch is turned off.	1. This is normal.

DETAILED TROUBLESHOOTING

The microprocessor is very complex, such that any error in the system results in a complete breakdown of the system. Open or shorted address, data, or control lines; their associated IC's; or a non-operating clock will all essentially show the same symptom (that is, when the unit is turned on, some or all of the LED segments will light, but nothing else happens). The following material gives you a systematic check of the Trainer circuitry to help you locate the problem. The material is divided into sections (which are listed below). If you know the section that the trouble is in, proceed to that section and start there. Otherwise, start at "Binary Data LED's."

- Binary Data LED's
- Clock
- Reset
- Address Lines
- Data Lines
- Control Lines
- Decoding

Binary Data LED's

If the +5-volt supply is operating, indicated by the LED1 next to the Power switch being lit, you can troubleshoot your Trainer without using test equipment.

Set the Power switch to On.

Cut a 14" length of yellow wire and remove 3/8" of insulation from each end. Refer to Pictorial 9-2 and insert one end of the wire into the LED connector block labeled 7, or to the block of an LED that you know does not work. Insert the other wire end into the +5 connector block. The LED directly above the connector block should light.

If the LED does not light:

- A. Visually inspect the LED's. The flat at the base of each LED should face the top of the circuit board.



- B. Unplug the line cord, remove IC1 from its socket, and plug the line cord back in.

With the indicated end of the yellow wire, one at a time, touch the eight circuit board plated-through holes shown. The eight LED's should light one at a time. If they do not, replace the LED that does not light.

- C. With pliers, flatten one end of the yellow wire.

Carefully insert the wire into pin 20 of socket IC1 and touch the other wire end to the indicated plated-through hole. The 0 LED should light. IF it does not light, check the IC socket pins and the circuit board foils to find out why +5 volts is not at pin 20 of the socket. Then remove the yellow wire.

- D. Unplug the line cord.

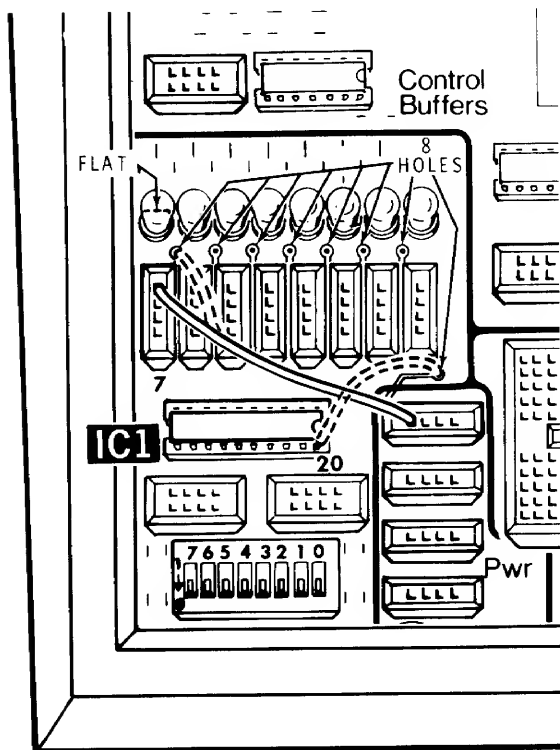
Be sure the pins of IC1 are straight and then properly reinstall the IC in its socket.

Reconnect the yellow wire to LED connector block 7 (or to the connector block of an LED that you know was not working) and the +5 connector block. The data LED should light. If it does not light, replace IC1.

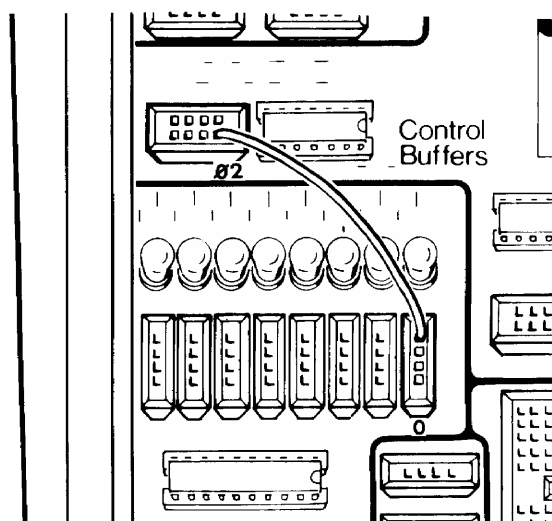
Clock

The simplest test to determine if the clock (IC19) is operating is to place a portable radio near the clock and tune the radio across the broadcast band. If the clock is operating, you will hear several "beat" signals. Unplug the Trainer's line cord and the beat signal will disappear if it is caused by the microprocessor clock.

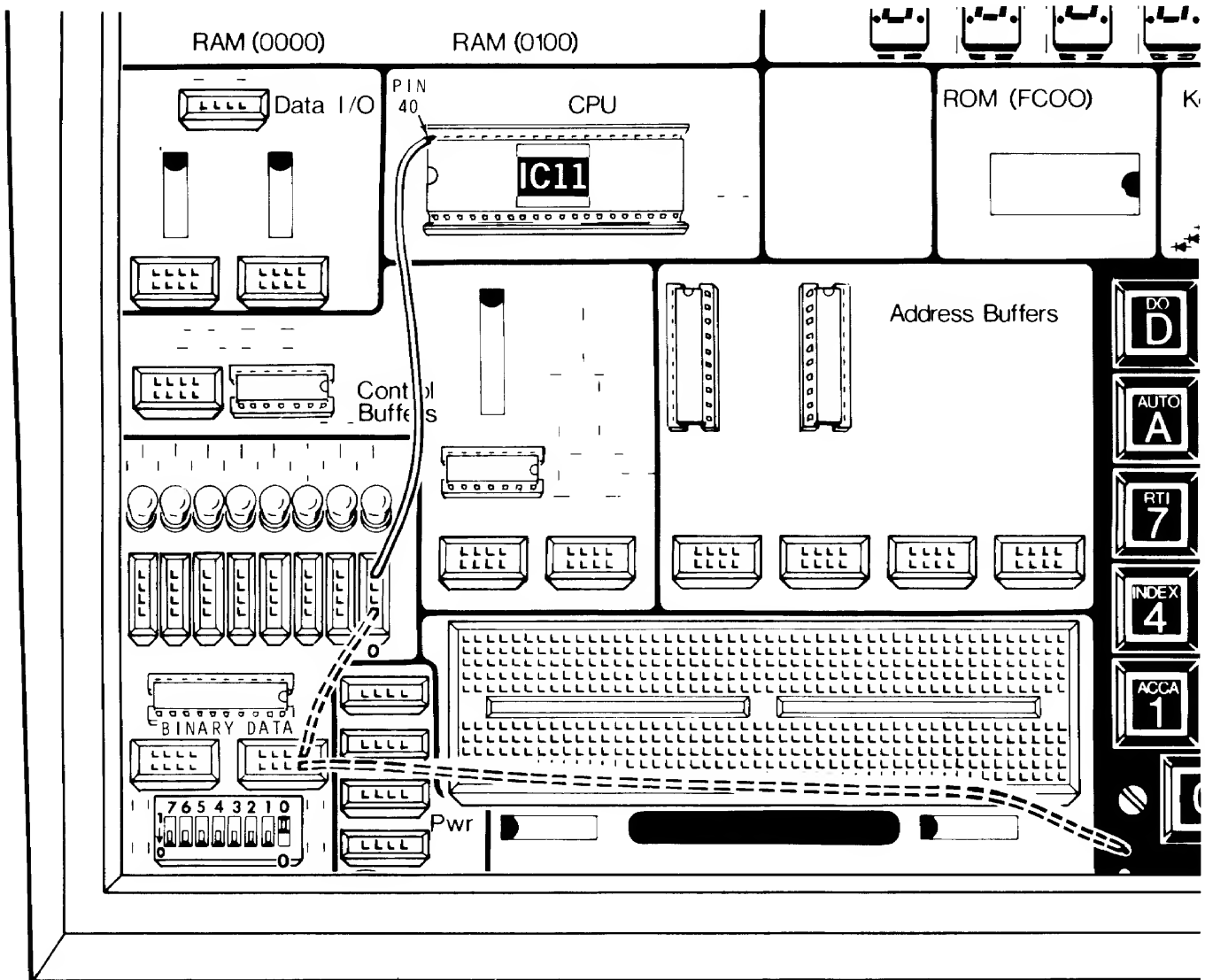
There are four different clock outputs used in the system (pins 7, 9, 13, and 15). Usually, the outputs of a clock that has failed will assume a logic 0 state. To test the clock, use a wire and connect a data LED (LED 2 through LED 9) to the clock's four outputs. If the clock is working, the LED will light but it will be noticeably dimmer than the same LED connected to +5 volts. This is due to the 50% duty cycle of the clock. See Pictorial 9-3.



PICTORIAL 9-2



PICTORIAL 9-3



PICTORIAL 9-4

Reset

Refer to Pictorial 9-4 and connect a wire from a data LED connector block to IC11 pin 40. The LED should light. While you hold the test lead on pin 40, press the Reset key. The LED should go out while the Reset key is pressed and come back on again when it is released. Then remove the wire.

To test the reset input, connect a wire from the indicated Binary Data connector block to LED connector block 0.

Set data switch 0 to logic 1.

Connect another wire from the indicated Binary Data connector block to the circuit board soldered connection just left of the 0 key. The lamp should stay on until you push the Reset key; then it will go out. It will come back on when you release the key.

Other effects of pushing the Reset key will be covered later after you check the address and data lines. Remove the two wires.



Test Wires

The following paragraphs instruct you how to make indicators for testing tri-state* devices. These are necessary for testing address and data lines in the following sections.

Unplug the line cord.

Refer to Pictorial 9-5 (Illustration Booklet, Page 9) and unsolder and disconnect the indicated lead of resistor R24 from the circuit board as shown.

Prepare a 2" yellow wire. Temporarily solder one end of the wire to the free lead of the resistor and plug the other end of the wire into connector block 6.

Prepare two 12" yellow wires. Remove 3/8" of insulation from both ends of each wire.

Insert one end of one wire into LED connector block 7.

Insert one end of the other wire into LED connector block 6.

In the following sections, these two wires will be referred to as test wire 7 and test wire 6. Be sure you reconnect and resolder the loose resistor lead after you locate and repair the problem.

Plug in the line cord. LED 7 should be on and the other data LED's should be off.

Address Lines

In checking the buffered address lines, you will look for two basic problems:

1. Lines that are shorted.
2. Lines that are not connected properly.

To perform these tests, you will tri-state the CPU. In this state, the address lines from the CPU and from the buffers are in a high impedance state. Therefore, any logic level can be put on these lines. Data input switches will apply test logic levels to the address lines, and data LED displays will serve as logic level indicators.

Prepare the following yellow wires. Cut them to the lengths shown and remove 3/8" of insulation from each end:

<u>WIRES</u>	<u>LENGTH</u>
3	4"
2	8"

Refer to Pictorial 9-5 and connect a 4" wire from the ground connector block to TSC.

Touch test wire 6 to the +5 connector block to test the LED. LED 6 should light. Touch test wire 6 to IC11 pin 39. LED 6 should again light; this indicates proper TSC voltage. If the LED does not light, proceed to "Control Lines" tests on Page 103.

Touch test wire 6 to IC7 pin 1 and then to IC8 pin 1. The LED should light both times. This indicates proper tri-state voltage. If the LED does not light, proceed to "Control Lines" tests on Page 103.

Touch test wire 7 to IC7 pin 19 and then to IC8 pin 19. LED 7 should go out both times; this indicates the correct voltage to tri-state the address buffers.

One after another, touch test wire 7 to each address output connector (A_0 - A_{15}). LED 7 will remain lit unless the line touched is shorted to logic 0. If the LED goes out, trace the foil pattern and look for a solder bridge. If this does not solve the problem, then remove the IC's connected to that line, one at a time, to check for a defective IC. CAUTION: Do not remove or install IC's with the line cord plugged in. If you remove a MOS IC, place it in the protective foam in which you received it. This will prevent possible damage from a static charge. (See Page 36 for instructions on how to handle MOS IC's.)

One after another, touch test wire 6 to each address output connector. The LED will remain unlit. If the LED should light, it indicates a short to logic 1. Use the same procedure as above to check for the cause of the problem.

* Registered Trademark, National Semiconductor

The next test is to make sure that none of the address lines are shorted together. To do this, you will put a logic 1 on one line and a logic 0 on the line beside it. If the two lines are shorted together, the logic 0 will cause both LED's to be off. You will also check to see that the address line is indeed connected to all the IC's where it should be.

Connect a 4" wire from the connector block of data switch 0 to the connector of data LED 0.

Connect a 4" wire from data switch 1 to LED 1.

Place the data 0 switch to logic 1 and the data 1 switch to logic 0.

Connect an 8" wire from data LED 0 to A_0 , and an 8" wire from data LED 1 to A_1 . LED 0 should be on and LED 1 should be out. If there is a short between lines A_0 and A_1 , both LED's will be out. If the LED's are out, check for solder bridges or defective IC's.

Connect test wire 6 to all the IC pins indicated in the following chart as being connected to A_0 (IC7 pin 12,

IC12 pin 24, etc.). The LED should light as each pin is touched. If it does not light, an open circuit exists between the address terminal and the pin being tested.

Move the wire that is at line A_1 to A_2 . Then move the wire at A_0 to A_1 .

As before, LED 0 should be on and LED 1 should be out. If both LED's are out, this time check for a short between lines A_1 and A_2 .

Connect test wire 6 to all the IC pins indicated in Test Chart A as being connected to A_1 . The LED should light as each pin is touched.

Continue moving the two wires towards A_{15} , one position at a time, and make the tests in the above three steps until all the address lines have been checked.



TEST CHART A

BUFFERED ADDRESS LINES	BUFFERS		ROM IC12	RAM IC14 IC15 IC16 IC17 PIN	DISPLAY LATCHES IC23 IC26 IC24 IC27 IC25 IC28 PIN	ADDRESS DECODING			
						IC2	IC3	IC20	IC22
	IC7 PIN	IC8 PIN	PIN			PIN	PIN	PIN	PIN
A ₀	12		24	4	1				
A ₁	14		23	3	2				
A ₂	16		22	2	3				
A ₃	18		21	1					
A ₄	9		20	15					15
A ₅	7		19	5					14
A ₆	5		18	6					13
A ₇	3		17	7					
A ₈		12	16					15	
A ₉		14	15					14	
A ₁₀		16	14				15		
A ₁₁		18	13				14		
A ₁₂		3	10				12		
A ₁₃		5				15			
A ₁₄		7				14			
A ₁₅		9				13			

To check the address lines between the CPU and the address connectors, it is necessary to remove the tri-state condition from the buffers and the CPU.

Be sure the line cord is unplugged.

Remove IC11 from its socket.

Remove IC5 and bend pin 11 out slightly. Then reinstall the IC so that pin 11 is not in the socket.

Remove all the previously installed wires except test wires 6 and 7.

Follow chart below and reconnect the wires.

WIRE	FROM	TO
4"	Data switch 0	LED 0
4"	Data switch 1	LED 1
4"	+5	\overline{BA}
8"	LED 0	A_0
8"	LED 1	A_1

Set data input switches 0 and 1 to logic 1.

Plug in the line cord.

Use test wire 6 and check for correct logic levels at IC7 and IC8. Pin 1 is logic 0 (LED 6 off) and pin 19 is logic 1 (LED 6 on).

Address lines A_0 and A_1 should be logic 1, indicated by LED 0 and LED 1 being lit. Remove the 12" test wire from LED 7 and insert one end in the GND connector.

Touch the free wire end to IC7 pin 8. The LED connected to A_0 should go out, while the LED connected to A_1 will remain lit. If both LED's go out, there is a short circuit between the A_0 and A_1 lines, between the CPU and the buffer inputs.

Follow Test Chart B to check all the address lines.



TEST CHART B

8" WIRE FROM LED 0 TO:	8" WIRE FROM LED 1 TO:	GND		TURNS OFF LED
		IC	PIN	
A ₀	A ₁	7	8	0
A ₂	A ₁	7	6	1
A ₂	A ₃	7	4	0
A ₄	A ₃	7	2	1
A ₄	A ₅	7	11	0
A ₆	A ₅	7	13	1
A ₆	A ₇	7	15	0
A ₈	A ₇	7	17	1
A ₈	A ₉	8	8	0
A ₁₀	A ₉	8	6	1
A ₁₀	A ₁₁	8	4	0
A ₁₂	A ₁₁	8	2	1
A ₁₂	A ₁₃	8	17	0
A ₁₄	A ₁₃	8	15	1
A ₁₄	A ₁₅	8	13	0
A ₁₄	A ₁₅	8	11	1

Properly replace IC11 and IC5 into their sockets.

Remove all the wires except the test wires.

Data Lines

To check data lines for opens and shorts, you will input data through the data buffers, alternate logic 0 and logic 1 on adjacent data lines, and then look for the correct data at the affected IC pins. To do this, you will need the following yellow wires. Cut them to the lengths specified and remove 3/8" of insulation from each end.

<u>WIRES</u>	<u>LENGTH</u>
3	8"
3	4"

If not already done, refer to Page 97 and prepare the two test wires as instructed there.

Refer to Pictorial 9-6 (Illustration Booklet, Page 9) and and install a 4" wire between GND and TSC to tri-state the CPU.

The data I/O buffers are bi-directional transceivers with the enable line to provide data to the output connectors.

Touch test wire 6 to pins 1 and 13 of IC9 and IC10. The LED should light, indicating that the buffers are in the right state.

Touch the test lead to each of the data connectors (D₀-D₇). The lamp should light at each terminal, indicating that the data lines are tri-stated and none of the data lines are shorted to ground. If the LED does not light, check both the terminal and the CPU sides of the data lines involved.

Install the following jumper wires.

<u>WIRE</u>	<u>FROM</u>	<u>TO</u>
4"	Data switch 0	LED 0
4"	Data switch 1	LED 1
8"	LED 0	D ₀
8"	LED 1	D ₁
8"	GND	RE

Set data switch 0 to logic 1.

Set data switch 1 to logic 0.

Refer to Test Chart C and touch test wire 6 to any IC pin to which line D₀ is connected. The LED should light. If the LED does not light, there is a short between the D₀ and D₁ lines. Visually check for shorts. Remove the IC's connected to the D₀ and D₁ lines, one at a time, to determine if a short exists in an IC. CAUTION: Do not remove or install IC's with the line cord plugged in. If a short is not indicated, test all pins to which line D₀ is connected by moving the data switch from logic 1 to logic 0 while you touch each pin with the test wire. If you do not obtain the correct results at all pins, check for an open circuit to the pin not showing the proper response. (NOTE: Line D₀ also goes to the display latches and is inserted at IC21 pins 9 and 10.)

Move the leads from LED 0 and LED 1 to buffer data connectors D₁ and D₂, and repeat the test for D₁. Continue this procedure until you have checked all the data lines.



TEST CHART C

	BUFFERS				CPU	ROM	RAM		KEYBOARD BUFFER IC13 PIN
	CONNECTOR SIDE		CPU SIDE				IC15 & IC17 PIN	IC14 & IC16 PIN	
	IC9 PIN	IC10 PIN	IC9 PIN	IC10 PIN					
D ₀		8		6	33	2	9		3
D ₁		9		5	32	3	10		5
D ₂		10		4	31	4	11		9
D ₃		11		3	30	5	12		7
D ₄	8		6		29	6		9	11
D ₅	9		5		28	7		10	13
D ₆	10		4		27	8		11	
D ₇	11		3		26	9		12	

Line D₀ is also applied to IC21 pins 9 and 10. The D₀ output, IC21 pin 8 and IC6 pin 2, is connected from the output of IC6 (pin 18) to pin 13 of IC23 through IC28.

Remove all the wires except the test wires.

Control Lines

If not already done, refer to Page 97 and prepare the two test wires as instructed there.

There are nine lines other than data, clock, and address that affect the operation of the CPU. Four lines are always logic 1, unless they are pulled low by an external connection. These are RESET, HALT, IRQ,

and $\overline{\text{NMI}}$. Reset has been checked earlier in this section and will be covered again later in greater detail. The three other lines are connected through noninverting buffers to the CPU. The connector and the associated CPU pin are therefore at the same logic level. To test these three lines, touch test wire 6 to the connector and then to the corresponding CPU pin. The LED should light at both locations. Then repeat this procedure with a wire installed from GND to the connector block associated with the line being checked. The LED should not light at either the connector or the CPU pin.

In the above test, if you fail to get the correct indication, check for open or shorted lines. Also, IC6 may be defective.

To check the five remaining control lines (R/\overline{W} , TSC, BA, VMA, and DBE) plus $\overline{VMA\phi 2}$, you will use Halt and TSC, which forces a given logic level to appear on these control lines. Refer to the following chart and connect a wire from ground to the designated connector, and check for the desired result by touching test wire 6 to the indicated connector or IC pin.

TOUCH GROUND WIRE TO CONNECTOR	DBE IC11 PIN 36	TSC IC11 PIN 39	R/\overline{W} IC11 PIN 34	VMA IC11 PIN 5	BA IC11 PIN 7	IC4 PINS 1, 4, 10	IC4 PIN 13	$\overline{VMA\phi 2}$	\overline{BA}	\overline{TSC}	R/\overline{W}
—	1	0	1	1	0	1	1	1 ¹	1	1	1 ²
HALT	0	0	0	0	1	0	0	1	0	1	1
\overline{TSC}	0	1	0	0	0	0	1	1	1	0	1

¹ The $\overline{VMA\phi 2}$ state does not appear to change. However, the LED will not be as brightly lit when the CPU is running as it is when the CPU is in the Halt or TSC states.

² Although the R/\overline{W} line changes, the output connector does not change because the buffer is tri-stated when R/\overline{W} is low.

RESET

Previous tests indicated that the logic level on this pin is correct.

When the Reset key is closed, reset goes low, VMA and BA are low, and R/\overline{W} is high. In addition, the CPU puts the first address of the reset sequence on the address line. This address is FFFE. Therefore, test all the address lines with test wire 6. They will all be logic 1 except for A_0 , which is logic 0.

Decoding

In this section, you will put various addresses on the lines and then refer to the "Decoding Chart" and look at logic levels at the decoding IC's to check their operation. In each case, $\overline{VMA\phi 2}$ must be logic 0 in order to provide the proper addressing.

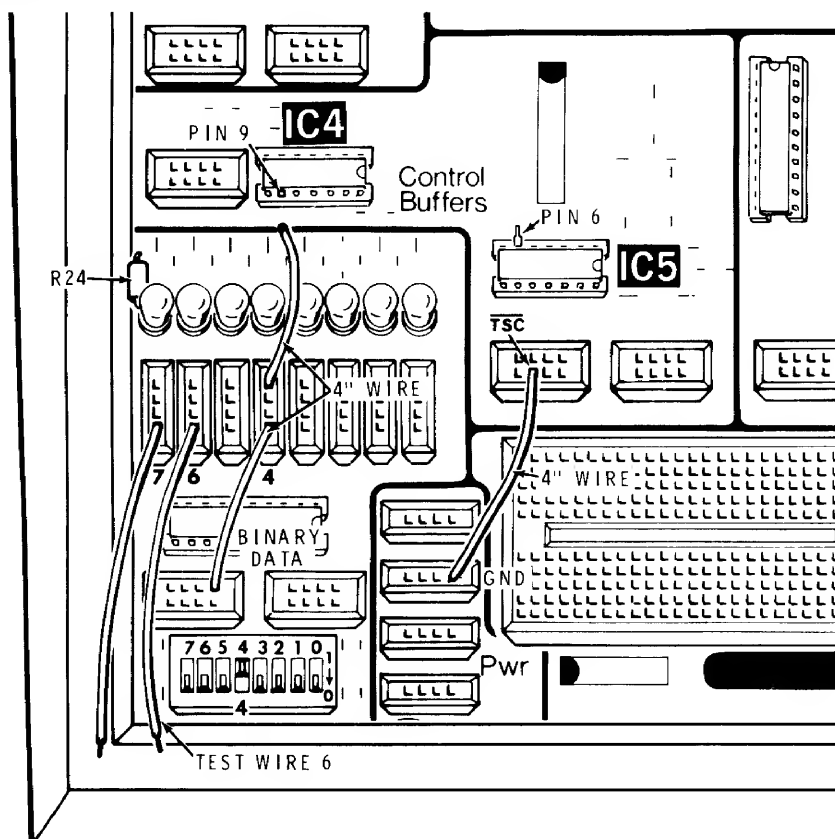
If not already done, refer to Page 97 and prepare the two test wires as instructed there.

$\overline{VMA\phi 2}$, for internal operation, is taken from the line connecting IC5 pin 6 to IC4 pin 9. To perform tests on the decoding section, you will need to pull this line to logic 0. To avoid damaging IC5 when you do this, refer to Pictorial 9-7, carefully remove IC5 from its socket, bend pin 6 out just far enough to clear the edge of the socket, and then reinstall the IC so that pin 6 is not in the socket.

Install one end of a 4" wire in LED 4. Insert the free end of this wire in the plated-through hole just below pin 9 of IC4. Temporarily solder this point on the bottom side of the circuit board.

Install a 4" wire from LED 4 to data switch 4. Data switch 4 will now determine the logic level of $\overline{VMA\phi 2}$, and LED4 will display the level. 1 is ON, 0 is OFF.

Install a 4" wire from GND to \overline{TSC} to tri-state the address lines so you can place an address on the lines.



PICTORIAL 9-7

In the following steps, refer to Pages 10 and 11 in the Illustration Booklet and use test wire 6 to check for proper address decoding. It is not necessary to go through the entire "Decoding Chart" unless the "End Result" is not correct. After you place an address on the address lines, check all the end results to make sure a problem does not exist, which results in more than the desired function being addressed. The logic level on the pins listed in the End Result column should be opposite of that indicated when the IC or function is not being addressed.

To address RAM 00 — Install a wire from A_{15} to GND. Then install wires from A_{15} to A_{14} , A_{14} to A_{13} etc., until lines A_8 through A_{15} are all connected together. To make sure $VMA\phi 2$ is doing its job, switch D4 between logic 1 and logic 0. The chart "End Result" should only be obtained when $VMA\phi 2$ is logic 0.

To address RAM 01 — Remove the wire installed between A_8 and A_9 for RAM 00 — Install a wire from A_8 to +5.

To address ROM FC — Remove the wire installed between A_9 and A_{10} , and install it between A_8 and A_9 . Move the wire from A_{15} to +5, and the wire from A_8 to GND instead of +5.

To address the keyboard C0-X — Remove the wire from between A_{14} and A_{13} , and install it between A_9 and A_{10} . Address lines A_3 through A_7 are "don't care" lines, so let them float. Install 4" wires from data switch 0, 1, and 2 to the corresponding LED terminals and 8" wires from the connectors to the corresponding address lines.

With the keyboard address on the lines, first look for the proper end result in the "Main Decoding Chart." If it is correct, proceed to the "Keyboard Column Address Decoding Chart."

In order to determine if a key is depressed, the monitor program causes the CPU to put the keyboard address on the line. Then it looks at the data lines to determine if a key is closed, which is indicated by the presence of a logic 0 on one of the affected data lines (D_0 through D_5). The eight high-order bits (C0 hex) are decoded and enable the keyboard buffer.

The three lower-order bits (3, 5, or 6 hex) place a logic 0 on one of the key columns. If a key is closed in the column address with a logic 0, a logic 0 will appear on the corresponding data line. Then you can tell which key is closed.

Place data switches 0, 1, and 2 in their logic 1 positions. The address lines to the key columns are all logic 1 and no key is actually addressed. Depress keys F, E, and D. All data lines should remain logic 1. If a line is logic 0, it indicates a shorted address line to the column of keys containing the depressed key.

Put the address for the right-hand column of keys (hex 3) on the three low bit address lines using the data switches. The LED will indicate that the address is correct. Connect the logic probe to each of the data output connectors, D_0 through D_5 . All the connections should be logic 1. If one of the data lines should be logic 0, a short to GND is indicated in the keyboard circuit. This could be caused by the key associated with the data line or it could be the row of three keys. For example, with the hex 3 address on the line, we find D_0 to be logic 0. The problem could be a short that only affects key F, or it could be a short affecting the row of keys D, E, and F. If you change the hex 3 portion of the address to either hex 5 or hex 6, and D_0 changes to logic 1, the short is only associated with key F. However, if the logic 0 remains, the problem is associated with the line to the entire row.

If the data lines are all logic 1 with no key depressed and a hex 3 address, depress Key F. Look at all the data lines while the key is depressed. Only D_0 should be logic 0. If, for example, lines D_0 and D_1 indicate logic 0, a short exists between keys F and C, or between the rows of keys D, E, F, and A, B, C. Again, to determine individual key versus rows of keys, change the column address to hex 5 and depress key E. If only D_0 is now logic 0, the problem exists between keys F and C. If D_0 and D_1 are logic 0, the problem is a short between the D, E, F and A, B, C rows or keys.

With the hex 3 address on the line, depress keys D and E. If data line D_0 goes to logic 0, a short is indicated between the column of key associated with the key depressed and the column containing the F key.

To address the display LED's CIX —. Remove the wire at A_8 - A_9 . Move the GND wire from A_8 to A_9 . Then connect a wire from A_8 to +5. Check for the proper end result indicated in the Main Decoding Chart. Move the 8" wires installed at A_0 , A_1 , and A_2 to A_4 , A_5 , and A_6 . Use data switches 0, 1, and 2 to apply the LED address as shown in the Display LED Chart. Test for the proper logic level at pin 14 of the addressed IC.

To address an LED segment CIXX, move the 8" wires from A_4 , A_5 , and A_6 to A_0 , A_1 , and A_2 respectively. Now use data switch 0, 1, and 2 to address the desired segment. Move the wire soldered to $\overline{VMA\phi 2}$ from LED4 to GND. Install 4" wires from data switches 3, 4, and 5 to LED connectors 3, 4, and 5. Install 8" wires from LED 3, 4, and 5 to address lines A_4 , A_5 , and A_6 . (NOTE: The data switches are one number from the corresponding address line so LED's 6 and 7 can still be used as logic level indicators.) Data switches 3, 4, and 5 can now be used to address the desired display LED.

The D_0 data line controls the state of the LED segment when the segment is addressed and $\overline{VMA\phi 2}$ is logic 0. If D_0 is logic 1, the segment will light and if D_0 is logic 0, the segment will be off. The D_0 data line is connected through IC21 and IC6 to the D data input (pin 13) of decoder latch IC's.

The "D₀ Logic Level Chart" shows the levels at the various connections on the D_0 segment control line. To control the logic level on the D_0 data line, connect an 8" wire from \overline{RE} to GND. Connect another 8" wire to the D_0 connector. The free end of this wire need not be connected to provide a logic 1, but it must be connected to GND to provide a logic 0 level on D_0 . To test this area, place the address for an LED and a segment on the low-order address lines, touch the output pin that should be affected with test wire 6, and then watch both the probe and the selected LED segment. If D_0 is logic 1, the segment should light and the logic probe should indicate logic 0. The reverse is true if D_0 is logic 0.

If you wish to check different LED's or segments, insert the D_0 input lead into the 1 Hz square wave connector. The address segment will turn on or off approximately every 1/2 second.

To test the latch action of the decoder latches, move the lead soldered to $\overline{VMA\phi 2}$ from GND to LED 6 and add a wire from LED connector 6 to data switch 6. If data switch 6 is logic 0, the addressed LED segment will follow the D_0 logic level. To check the latching action, move data switch 6 from 0 to 1 while the addressed LED segment is either on or off. The segment should remain in the state it is in when $\overline{VMA\phi 2}$ is moved to logic 1.

Remove the wires from your Trainer, properly replace IC5, and then reconnect and resolder the free lead of resistor R24.

SPECIFICATIONS

CPU (Central Processing Unit)	8-bit parallel, NMOS, bus oriented 6800.
ROM (Read Only Memory)	NMOS, 1024 bytes.
RAM (Random Access Memory)	NMOS, 256 bytes (plus sockets for additional 256 bytes).
Clock Frequency	500 kHz (approximately).
Display	Six 7-segment LED digits.
Keyboard	Hexadecimal (0-F and Reset). 1 through F are dual-function keys and also enter commands.
Input Switches	Eight miniature switches in a dual-in-line package. (For experiments.)
LED Monitor Lights	Eight red LED's with separate input terminals. (For experiments.) +5 volts at 1.5 amperes (.5A available for breadboard at output terminal.)
Power Supplies	+12 volts, and -12 volts at 50 milliamperes at output terminals.
Power Requirements	105-130 volts or 210-260 volts rms, 50-60 Hz. 30 watts maximum.
Fuse	3/8-ampere, slow-blow.
Dimensions	12-1/8" wide × 11-3/4" deep × 3-1/2" high. (30.8 × 29.8 × 8.9 cm.)
Net Weight	4 lbs (1.8 kg).

The Heath Company reserves the right to discontinue products and to change specifications at any time without incurring any obligation to incorporate new features in products previously sold.

THEORY OF OPERATION

As you read this section, refer to the Block Diagram (Illustration Booklet, Page 12) and the Schematic (fold-in).

The operation of the CPU (microprocessor, IC11) is very complex. Therefore, this section of the Manual will not discuss the internal operation of the CPU, but will discuss how the sections of circuitry in your Trainer operate together. For information concerning the CPU, refer to Motorola's M6800 application manual.

Many lines are connected to front panel connectors, as shown on the Schematic. Some are buffered and some are not. Most of these connections and their buffers will not be mentioned in the following paragraphs.

The Reset key is connected to the clock (IC19) which produces a proper reset pulse. This pulse is applied through tri-state buffer IC6 to the reset input (pin 40) of the CPU.

Two non-overlapping outputs are connected from the clock to the CPU. The memory $\phi 2$ output is used for internal timing and is connected through IC4 to the DBE input (pin 36) of the CPU.

The VMA line from the CPU is buffered by IC6 and then NAnDED by IC5B with memory $\phi 2$ to produce $\overline{\text{VMA}}\phi 2$. This signal is then applied to the address decoding circuits.

The CPU $\text{R}/\overline{\text{W}}$ line is coupled through IC4 to the $\text{R}/\overline{\text{W}}$ inputs of RAM.

The input signal to TSC is applied through inverter IC5C. TSC is normally logic 0 and is connected through IC4 to the input of IC5A and to $\overline{\text{G}}$ of the address buffers, IC7 and IC8. Line BA is normally connected through IC5D to the control line of the TSC portion of IC4. The output of IC5A is logic 1 and is connected to the control lines in IC4 for $\text{R}/\overline{\text{W}}$, DBE, and $\overline{\text{VMA}}\phi 2$; keeping these sections enabled. The output of IC5A is also connected to an enable input of the address buffers.

If TSC is pulled to logic 1, the input to IC5A and $\overline{\text{G}}$ on the address buffers also become logic 1. The output of IC5A and, therefore, the inputs to address enable and

the control lines for the other three sections of IC4 become logic 0. The address, $\text{R}/\overline{\text{W}}$, DBE, and $\overline{\text{VMA}}\phi 2$ buffers are all tri-stated. In this state, DBE is held at logic 0 by a pull-down resistor and the other three lines are held at logic 1 by pull-up resistors. When BA goes to logic 0, the TSC section of IC4 is tri-stated, TSC does not control the output, and the output is held at logic 1 by a pull-up resistor which tri-states the address buffers $\text{R}/\overline{\text{W}}$, DBE, and $\overline{\text{VMA}}\phi 2$ as described above.

The address lines are buffered by IC7 and IC8. The buffers have active outputs or are tri-stated as previously described.

The eight high-order address lines are connected to the address decoding IC's; $\overline{\text{VMA}}\phi 2$ is also applied to the decoding section. This line must be logic 0 to obtain proper decoding. With the high-order byte 00 on the lines, a logic 0 is placed on CE for IC14 and IC15, and its 256 bytes of RAM memory may be addressed by lines A_0 - A_7 . $\text{R}/\overline{\text{W}}$ from the CPU determines if information is to be stored into or read from the RAM.

High-order byte 01 does the same thing for the optional 256 bytes of RAM at that address.

With the high-order byte FC, FD, FE, FF; the address decoder places a logic 0 in $\overline{\text{CS}}1$ of the ROM. Address lines A_{10} , A_{11} , and A_{12} place logic 1 on $\text{CS}0$, $\text{CS}2$, and $\text{CS}3$; and lines A_0 through A_9 can address the 1024 bytes of read only memory.

Buffer IC13 is normally in its tri-state condition. When the high-order address byte C0 is decoded, a logic 0 is placed on its control lines to enable it. Address lines A_0 , A_1 , and A_2 apply a logic 0 to one of the key columns and logic 1 to the other two columns. If a key is closed in the column with Logic 0 on it, a logic 0 is placed on the data line for the row of keys. Which key is closed is determined by the monitor program by knowing the address that is on the line and which data line is 0. The diodes in series with the three address lines serve as buffers to prevent two adjacent keys from accidentally changing the address due to the lines being shorted together.



When high-order address byte C1 is decoded, the output of the decoder places a logic 0 on the D input of IC22. IC22 is a 4 to 10 line decoder. If a BCD number from 0 through 10 is placed on the inputs, the output line corresponding to that number will be logic 0. Output lines 1 through 6 are connected to the enable inputs of the six display latch drivers, IC23 through IC28. If the D input to IC22 (which is BCD equivalent of 8) is high, the BCD input will always be greater than 8 and the output lines actually in use cannot be decoded. With the D input held low, the BCD information supplied to the other three inputs will be 0 through 7. These three inputs are connected to address lines A₄, A₅, and A₆ and will determine which output line will be logic 0 by their logic levels. A hex 6 or BCD 110 on lines A₆, A₅, and A₄ will cause the enable line for the left-most latch driver (IC23 and DISPLAY LED H) to be logic 0. Hex 1 or BCD 001 enables IC28 and DISPLAY LED C.

Address lines A₀, A₁, and A₂ are connected to the latch select inputs of all six latch drivers. The BCD code on these lines (hex 0 through 7), is decoded in the enabled IC and results in the corresponding output line following the logic level on the D input of that IC. Each of the output lines is connected through one of seven segments of display LED or decimal point, and a current-limiting resistor, to +5 V. If the D input is logic 0 the addressed output line will be 0 also, and a corresponding segment will be lit. If D is logic 1, the output line is 1 and the segment will be out. The D₀ data line is inverted by IC21C and applied to the latch driver D inputs through IC6. Therefore, if line D₀ is logic 1, the D input is logic 0 and the addressed segment will be lit. If D₀ is logic 0, the addressed segment is off. The status of the output lines and LED segment, as determined by the address and D₀ logic level, is then latched when the enable line returns to logic 1.

The data lines are connected directly to the various devices in the system. Data buffers IC9 and IC10 are bus transceivers. They are wired to normally provide output from the data lines to the data terminals. Connecting \overline{RE} to logic 0 reverses the input output pins so you can input data from the connectors.

BINARY DATA SECTION

The eight section data switch has one side of all switches connected to ground. The other side of each section has a 4700 ohm pull-up resistor to the switched 5 V power supply. The connectors above the

switch provide convenient connection for two wires to each switch section. With a switch in the lower (closed) position, the associated terminal will provide a logic 0 level (ground). In the up (open) position the level will be logic 1. The switch sections are numbered 0 through 7 from right to left. The eight connectors numbered 0 through 7 are inputs to the non-inverting buffer IC1. An 8200 ohm pull-down resistor is connected through each input terminal to ground to hold the input at logic 0 when no connection is made to the terminals. Each buffer output is connected through an LED and a 180 ohm current-limiting resistor to ground. When the inputs to the buffer are logic 0, the outputs are also 0 and the LED is off. When the input rises to logic 1, the output also rises to logic 1 and lights the LED.

POWER SUPPLIES

The voltage from one of the center-tapped secondary windings (green leads) of power transformer T1 is rectified by diodes D1 and D2, filtered by capacitor C1, and regulated by IC31 to produce the +5-volt DC supply. With switch SW1 in the On position, +5 volt is supplied throughout the system. When SW1 is in the Standby position, +5 volt is not supplied to the display LED's, data switches, or the +5 V connector block.

The other center-tapped secondary winding (red) is rectified by diodes D3 and D5, filtered by C7, and regulated by IC30 to provide a -12-volt supply. This same winding is rectified by diodes D4 and D6, filtered by C6, and regulated by IC29 to provide a +12-volt supply. These two supplies are provided for bread-boarding and are not connected in the system. They are available at the appropriate connector blocks only when switch SW1 is in the On position.

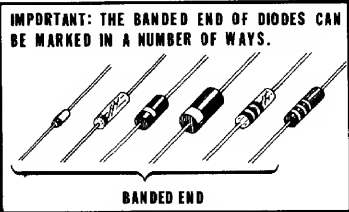
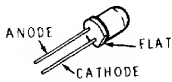
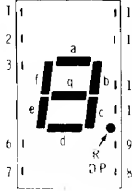
SQUARE WAVE OUTPUTS

The AC voltage at the anode of diode D6 is coupled by R5 and R6 to a section of voltage comparator IC18. Diode D10 keeps the AC voltage from driving the input negative with respect to ground. This section of the comparator is a zero-crossing detector to provide a symmetrical TTL compatible square wave that is in sync with the AC line.

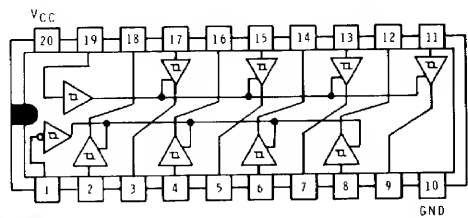
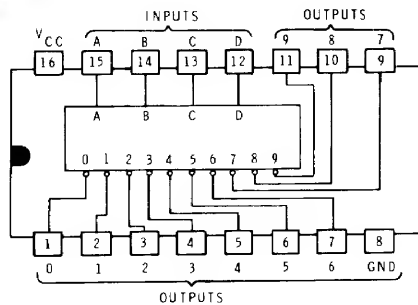
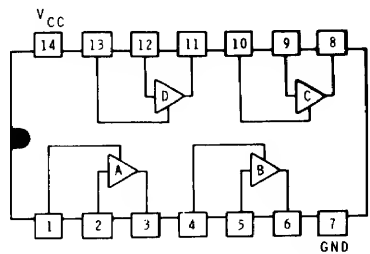
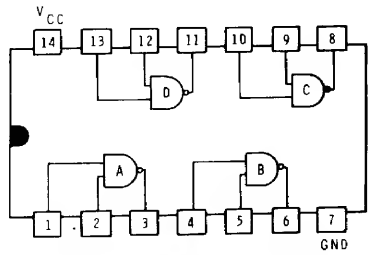
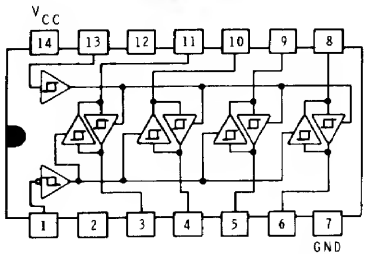
A second section of IC18 is used as an oscillator to produce a TTL compatible square wave at approximately 1 Hz. The symmetry and frequency of the square wave are determined by C13, R13, and R14.

SEMICONDUCTOR IDENTIFICATION CHARTS

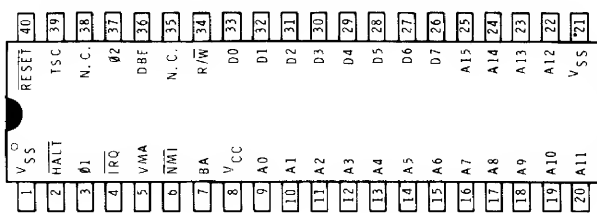
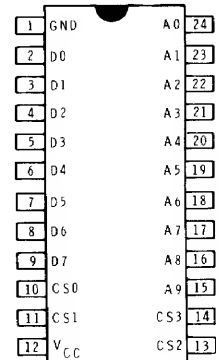
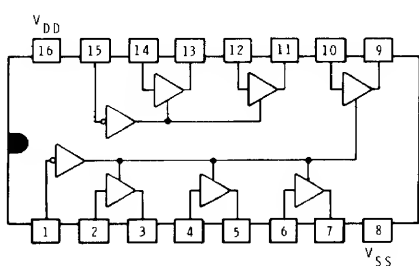
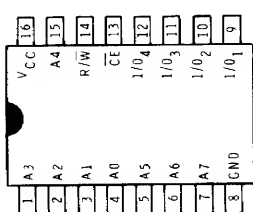
DIODES

COMPONENT	HEATH PART NUMBER	MAY BE REPLACED WITH	IDENTIFICATION
D1, D2	57-42	3A1	
D3, D4, D5, D6	57-65	1N4002	
D7, D8, D9, D10	56-56	1N4149	
LED1	412-611		
LED2, LED3, LED4, LED5, LED6, LED7, LED8, LED9	412-616	FLV117	
H, I, N, Z, V, C	411-831	TIL312	<p>TOP VIEW</p>  <p>PIN</p> <ul style="list-style-type: none"> 1... SEGMENT a 2... SEGMENT f 3... COMMON ANODE 4 NOT USED 5 NOT USED 6... NOT USED 7... SEGMENT e 8... SEGMENT d 9... RIGHT DECIMAL 10... SEGMENT c 11... SEGMENT g 12 NOT USED 13... SEGMENT b 14... COMMON ANODE <p>NOTE: PINS 3 AND 14 ARE INTERNALLY CONNECTED TOGETHER.</p>

INTEGRATED CIRCUITS

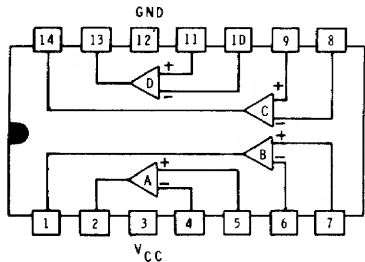
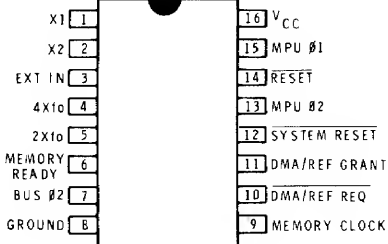
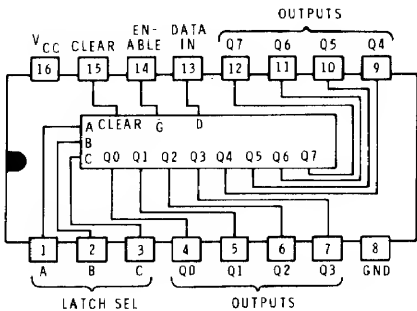
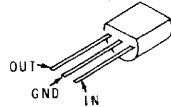
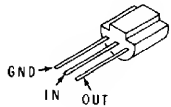

COMPONENT	HEATH PART NUMBER	MAY BE REPLACED WITH	IDENTIFICATION
IC1, IC6, IC7, IC8	443-824	74LS241	
IC2, IC3 IC20, IC22	443-807	74LS42	
IC4	443-717	74126	
IC5, IC21	443-26	74S00	
IC9, IC10	443-839	74LS243	

Integrated Circuits, Cont'd.

COMPONENT	HEATH PART NUMBER	MAY BE REPLACED WITH	IDENTIFICATION
IC11	443-827	MC6800P	
IC12	444-17	MCM6830A *	
IC13	443-720	40097	
IC14, IC15, IC16, IC17	443-721	2112-2	

* Must be mask programmed from the listing in this Manual.

Integrated Circuits Cont'd.

COMPONENT	HEATH PART NUMBER	MAY BE REPLACED WITH	IDENTIFICATION
IC18	442-616	LM3302N, LM2901N, or μ A775 (selected)	
IC19	443-840	MC6875	
IC23, IC24, IC25, IC26, IC27, IC28	443-804	74LS259	
IC29	442-644	LM78L12	
IC30	442-646	LM79L12AC	
IC31	442-30	μ A309K	

FOR PARTS REQUESTS ONLY

- Be sure to follow instructions carefully.
- Use a separate letter for all correspondence.
- Please allow 10 - 14 days for mail delivery time.

DO NOT WRITE IN THIS SPACE**INSTRUCTIONS**

- Please print all information requested.
- Be sure you list the correct **HEATH** part number exactly as it appears in the parts list.
- If you wish to prepay your order, mail this card and your payment in an envelope. Be sure to include 10% (25¢ minimum, \$3.50 maximum) for insurance, shipping and handling. Michigan residents add 4% tax.

Total enclosed \$_____

- If you prefer COD shipment, check the COD box and mail this form. COD ☐

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

The information requested in the next two lines is not required when purchasing nonwarranty replacement parts, but it can help us provide you with better products in the future.

Model # _____ Invoice # _____

Date Purchased _____ Location Purchased _____

LIST HEATH PART NUMBER	QTY.	PRICE EACH	TOTAL PRICE

TOTAL FOR PARTS

HANDLING AND SHIPPING

MICHIGAN RESIDENTS ADD 4% TAX

TOTAL AMOUNT OF ORDER

SEND TO: **HEATH COMPANY**
BENTON HARBOR
MICHIGAN 49022
ATTN: PARTS REPLACEMENT

Phone (Replacement parts only): 616 982-3571

THIS FORM IS FOR U.S. CUSTOMERS ONLY
OVERSEAS CUSTOMERS SEE YOUR DISTRIBUTOR

FOR PARTS REQUESTS ONLY

- Be sure to follow instructions carefully.
- Use a separate letter for all correspondence.
- Please allow 10 - 14 days for mail delivery time.

DO NOT WRITE IN THIS SPACE**INSTRUCTIONS**

- Please print all information requested.
- Be sure you list the correct **HEATH** part number exactly as it appears in the parts list.
- If you wish to prepay your order, mail this card and your payment in an envelope. Be sure to include 10% (25¢ minimum, \$3.50 maximum) for insurance, shipping and handling. Michigan residents add 4% tax.

Total enclosed \$_____

- If you prefer COD shipment, check the COD box and mail this form. COD ☐

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

The information requested in the next two lines is not required when purchasing nonwarranty replacement parts, but it can help us provide you with better products in the future.

Model # _____ Invoice # _____

Date Purchased _____ Location Purchased _____

LIST HEATH PART NUMBER	QTY.	PRICE EACH	TOTAL PRICE

TOTAL FOR PARTS

HANDLING AND SHIPPING

MICHIGAN RESIDENTS ADD 4% TAX

TOTAL AMOUNT OF ORDER

SEND TO: **HEATH COMPANY**
BENTON HARBOR
MICHIGAN 49022
ATTN: PARTS REPLACEMENT

Phone (Replacement parts only): 616 982-3571

THIS FORM IS FOR U.S. CUSTOMERS ONLY
OVERSEAS CUSTOMERS SEE YOUR DISTRIBUTOR

CUT ALONG DOTTED LINE

CUSTOMER SERVICE

REPLACEMENT PARTS

Please provide complete information when you request replacements from either the factory or Heath Electronic Centers. Be certain to include the **HEATH** part number exactly as it appears in the parts list.

ORDERING FROM THE FACTORY

Print all of the information requested on the parts order form furnished with this product and mail it to Heath. For telephone orders (parts only) dial 616 982-3571. If you are unable to locate an order form, write us a letter or card including:

- Heath part number.
- Model number.
- Date of purchase.
- Location purchased or invoice number.
- Nature of the defect.
- Your payment or authorization for COD shipment of parts not covered by warranty.

Mail letters to: Heath Company
Benton Harbor
MI 49022
Attn: Parts Replacement

Retain original parts until you receive replacements. Parts that should be returned to the factory will be listed on your packing slip.

OBTAINING REPLACEMENTS FROM HEATH ELECTRONIC CENTERS

For your convenience, "over the counter" replacement parts are available from the Heath Electronic Centers listed in your catalog. Be sure to bring in the original part and purchase invoice when you request a warranty replacement from a Heath Electronic Center.

TECHNICAL CONSULTATION

Need help with your kit? — Self-Service? — Construction? — Operation? — Call or write for assistance. you'll find our Technical Consultants eager to help with just about any technical problem except "customizing" for unique applications.

The effectiveness of our consultation service depends on the information you furnish. Be sure to tell us:

- The Model number and Series number from the blue and white label.
- The date of purchase.
- An exact description of the difficulty.
- Everything you have done in attempting to correct the problem.

Also include switch positions, connections to other units, operating procedures, voltage readings, and any other information you think might be helpful.

Please do not send parts for testing, unless this is specifically requested by our Consultants.

Hints: Telephone traffic is lightest at midweek — please be sure your Manual and notes are on hand when you call.

Heathkit Electronic Center facilities are also available for telephone or "walk-in" personal assistance.

REPAIR SERVICE

Service facilities are available, if they are needed, to repair your completed kit. (Kits that have been modified, soldered with paste flux or acid core solder, cannot be accepted for repair.)

If it is convenient, personally deliver your kit to a Heathkit Electronic Center. For warranty parts replacement, supply a copy of the invoice or sales slip.

If you prefer to ship your kit to the factory, attach a letter containing the following information directly to the unit:

- Your name and address.
- Date of purchase and invoice number.
- Copies of all correspondence relevant to the service of the kit.
- A brief description of the difficulty.
- Authorization to return your kit COD for the service and shipping charges. (This will reduce the possibility of delay.)

Check the equipment to see that all screws and parts are secured. (Do not include any wooden cabinets or color television picture tubes, as these are easily damaged in shipment. Do not include the kit Manual.) Place the equipment in a strong carton with at least **THREE INCHES** of *resilient* packing material (shredded paper, excelsior, etc.) on all sides. Use additional packing material where there are protrusions (control sticks, large knobs, etc.). If the unit weighs over 15 lbs., place this carton in another one with 3/4" of packing material between the two.

Seal the carton with reinforced gummed tape, tie it with a strong cord, and mark it "Fragile" on at least two sides. Remember, the carrier will not accept liability for shipping damage if the unit is insufficiently packed. Ship by prepaid express, United Parcel Service, or insured Parcel Post to:

Heath Company
Service Department
Benton Harbor, Michigan 49022



HEATH COMPANY • BENTON HARBOR, MICHIGAN
THE WORLD'S FINEST ELECTRONIC EQUIPMENT IN KIT FORM

LITHO IN U.S.A.

HEATH PARTS PRICE LIST
ET -3400 ECL 09

10/20/80

PAGE 1 OF 1

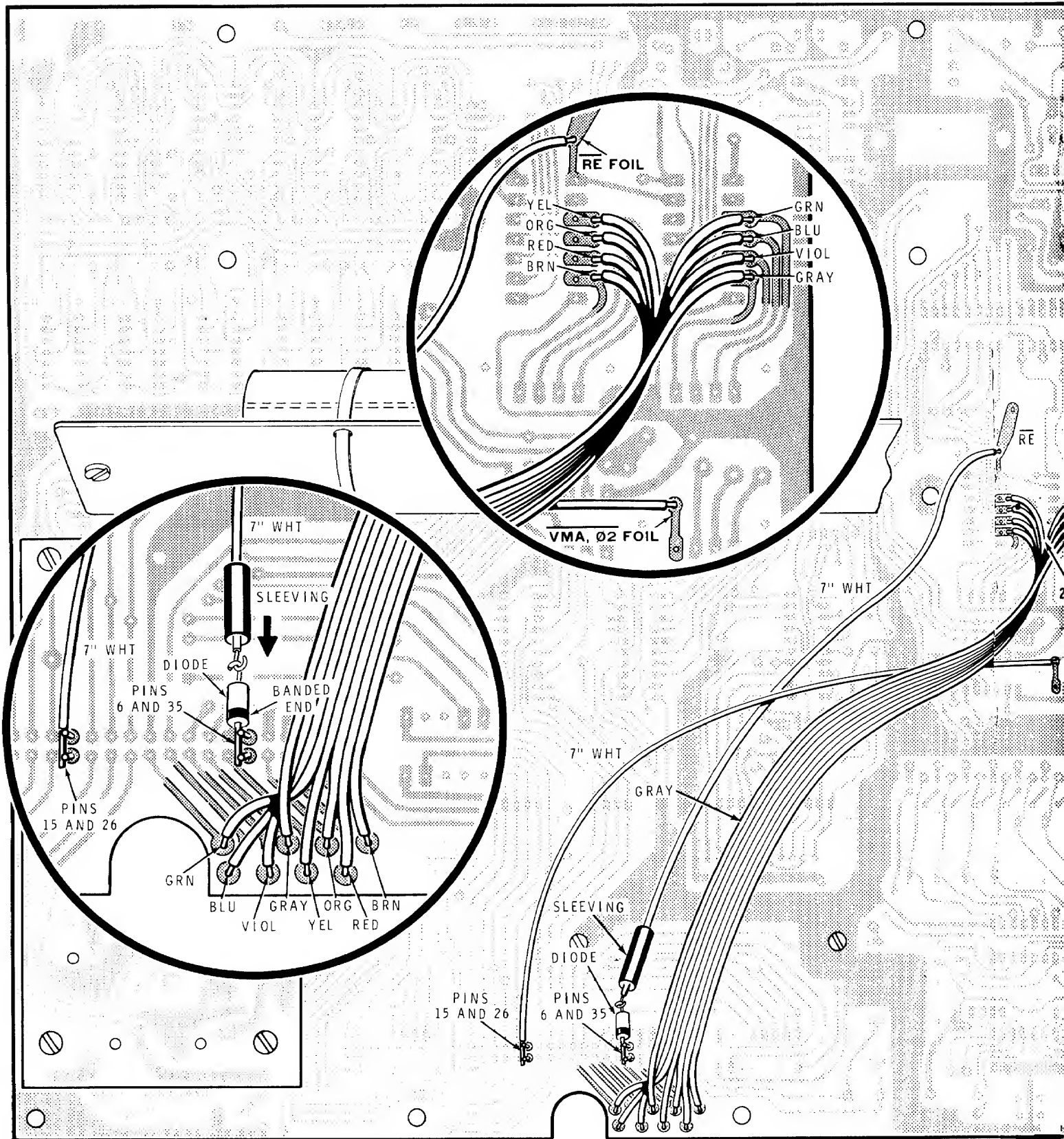
KEEP THIS PARTS LIST WITH YOUR MANUAL AND USE THE PRICES SHOWN BELOW WHEN ORDERING PARTS. THESE PRICES ARE SUBJECT TO CHANGE WITHOUT NOTICE.

THE PRICES SHOWN ON THE "HEATH PARTS PRICE LIST" APPLY ONLY ON PURCHASES FROM THE HEATH COMPANY WHERE SHIPMENT IS TO A U.S.A. DESTINATION. ADD 10% (MINIMUM 25 CENTS) TO THE PRICE WHEN ORDERING (MICHIGAN RESIDENTS ADD 4% SALES TAX) TO COVER INSURANCE, POSTAGE, AND HANDLING. OUTSIDE THE U.S.A., PARTS AND SERVICE ARE AVAILABLE FROM YOUR LOCAL HEATHKIT SOURCE AND WILL REFLECT ADDITIONAL TRANSPORTATION, TAXES, DUTIES, AND RATES OF EXCHANGE.

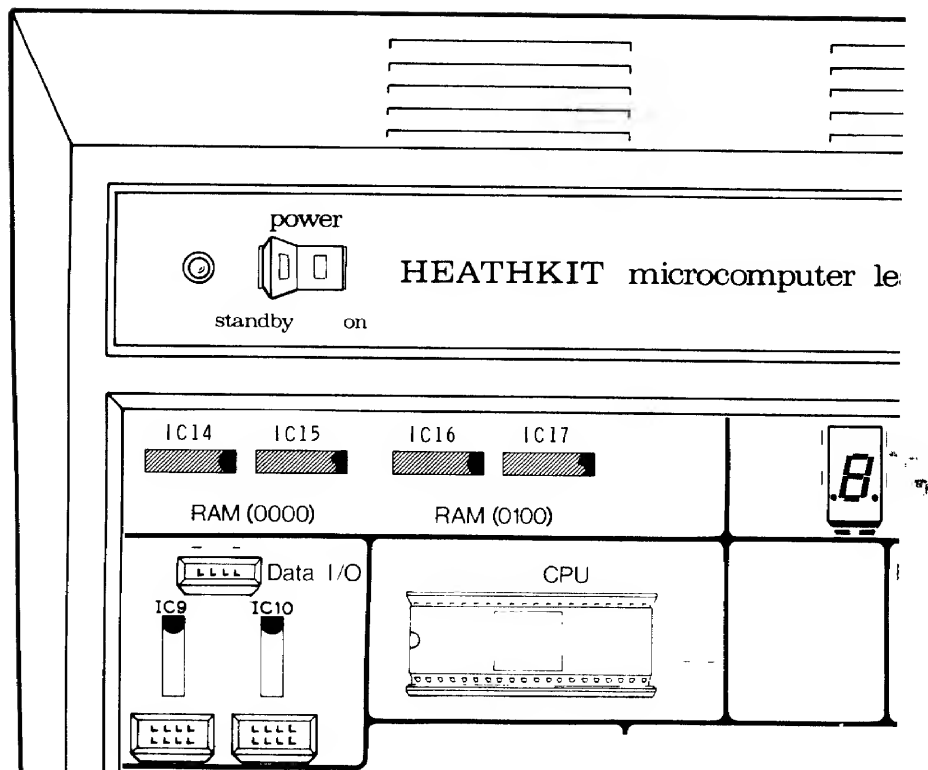
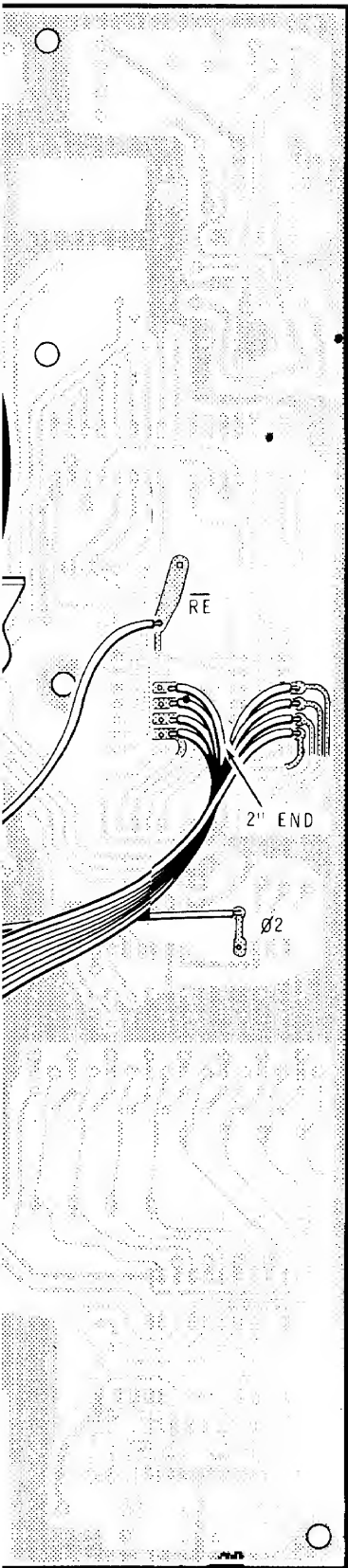
ADDITIONAL 3 FT ROLLS OF SOLDER, #331-6, CAN BE ORDERED FOR 25 CENTS EACH.

PART NUMBER	PRICE	*	PART NUMBER	PRICE	*	PART NUMBER	PRICE	*	PART NUMBER	PRICE	*
6-104-12	.30	*	255-23	.20	*	443-824	4.60	*			*
6-122-12	.30	*	259-1	.05	*	443-827	24.20	*			*
6-151-12	.30	*	259-22	.05	*	443-839	4.45	*			*
6-153-12	.30	*	260-56	.15	*	443-840	9.75	*			*
6-154-12	.30	*	260-700	.25	*	444-17	14.55	*			*
6-181-12	.30	*	261-34	.20	*	462-1023	.45	*			*
6-224-12	.30	*	262-8	.05	*	475-12	.10	*			*
6-273-12	.30	*	344-51	.05@	*	490-111	.15	*			*
6-471-12	.30	*	344-52	.05@	*			*			*
6-472-12	.30	*	344-53	.05@	*			*			*
6-680	.25	*	344-54	.05@	*			*			*
6-822-12	.30	*	344-71	.05@	*			*			*
6-822-12	.30	*	344-73	.05@	*			*			*
6-824-12	.30	*	344-74	.05@	*			*			*
20-102	.50	*	344-99	.05@	*			*			*
21-176	.35	*	345-1	.15@	*			*			*
23-200	.95	*	346-1	.10@	*			*			*
25-220	1.30	*	352-13	.25	*			*			*
25-221	1.00	*	354-7	.15	*			*			*
25-241	1.55	*	390-1255	.15	*			*			*
25-272	2.00	*	390-1390	.15	*			*			*
27-85	.50	*	390-1391	1.10	*			*			*
34-920	11.50	*	390-1395	.60	*			*			*
56-56	.60	*	390-1404	.45	*			*			*
57-43	.75	*	411-831	2.80	*			*			*
57-65	.50	*	412-616	.40	*			*			*
60-34	2.60	*	412-640	.90	*			*			*
60-621	3.85	*	421-42	.65	*			*			*
64-839	1.00	*	431-2	.15	*			*			*
73-4	.05	*	431-86	.15	*			*			*
75-718	.45	*	432-874	11.40	*			*			*
75-724	.20	*	432-875	11.50	*			*			*
75-784	.40	*	432-921	.40	*			*			*
75-788	.30	*	432-973	.80	*			*			*
85-2010-1	2.95	*	434-253	1.60	*			*			*
85-2033-3	28.45	*	434-298	.30	*			*			*
89-49	1.20	*	434-299	.30	*			*			*
92-611	2.60	*	434-307	.95	*			*			*
92-612	3.90	*	434-311	.90	*			*			*
204-2291	1.65	*	434-336	1.20	*			*			*
250-32	.05	*	442-30	4.55	*			*			*
250-56	.05	*	442-616	2.70	*			*			*
250-138	.05	*	442-644	1.10	*			*			*
250-162	.05	*	442-646	1.30	*			*			*
250-163	.05	*	443-26	1.05	*			*			*
250-473	.05	*	443-717	1.30	*			*			*
250-559	.10	*	443-720	1.25	*			*			*
250-1137	.15	*	443-721	7.20	*			*			*
252-3	.05	*	443-804	3.05	*			*			*
254-1	.05	*	443-807	1.50	*			*			*

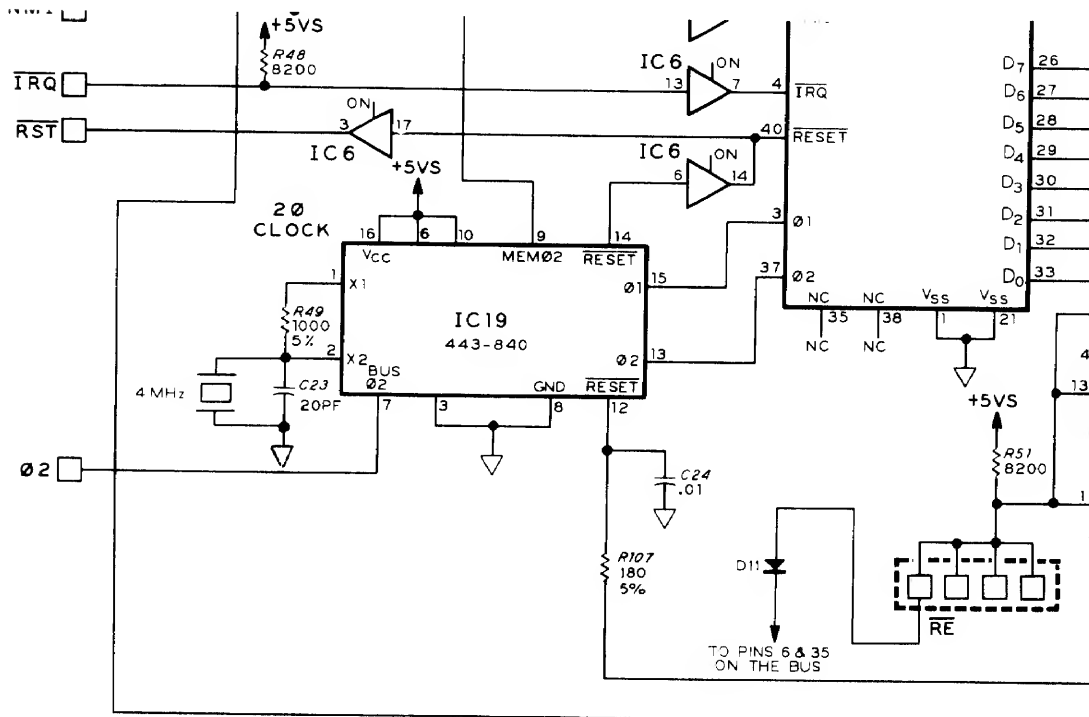
***** WRITE HEATH COMPANY FOR PRICE INFORMATION.
@ PRICE PER FOOT.



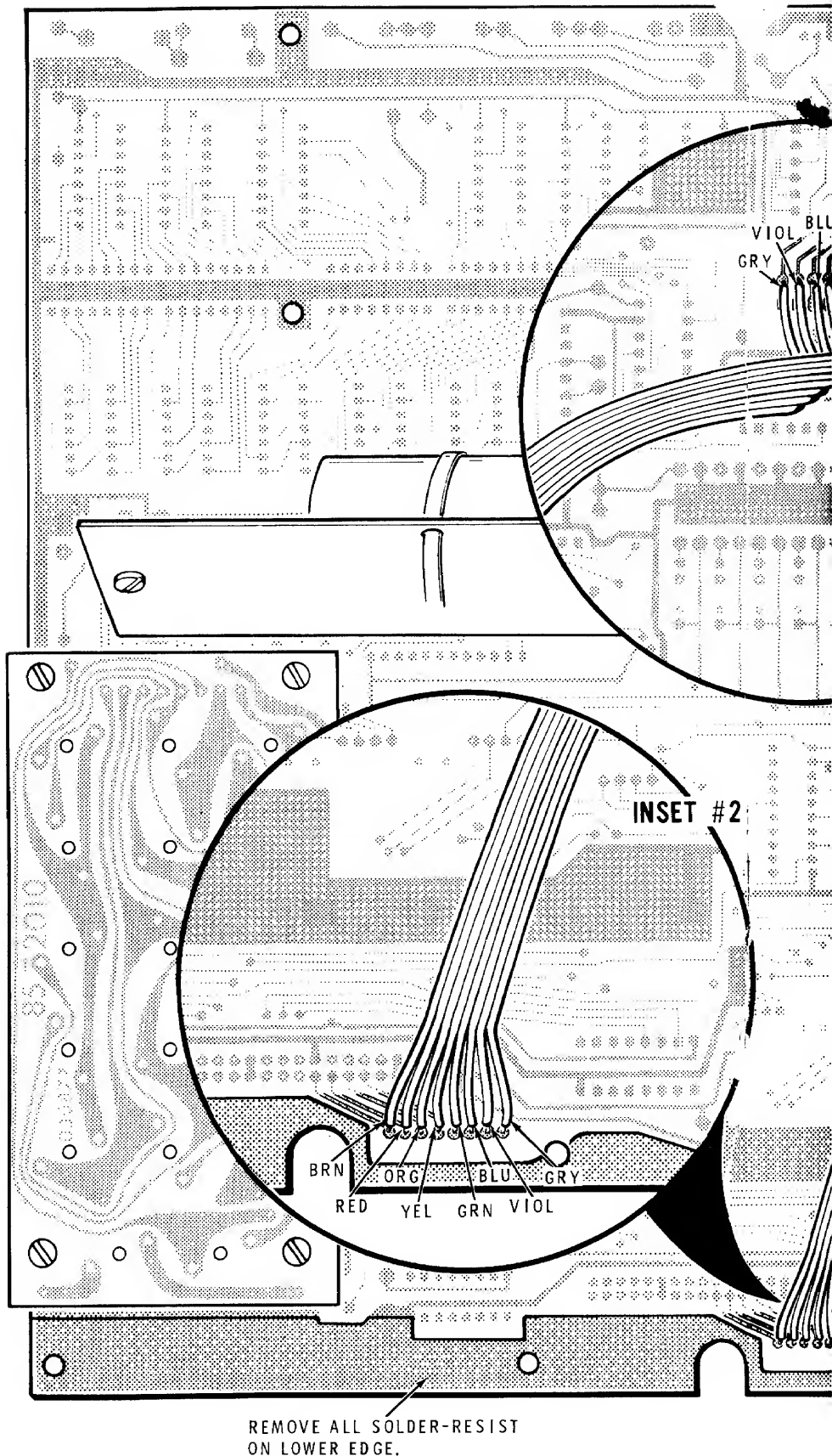
PICTORIAL 1-6



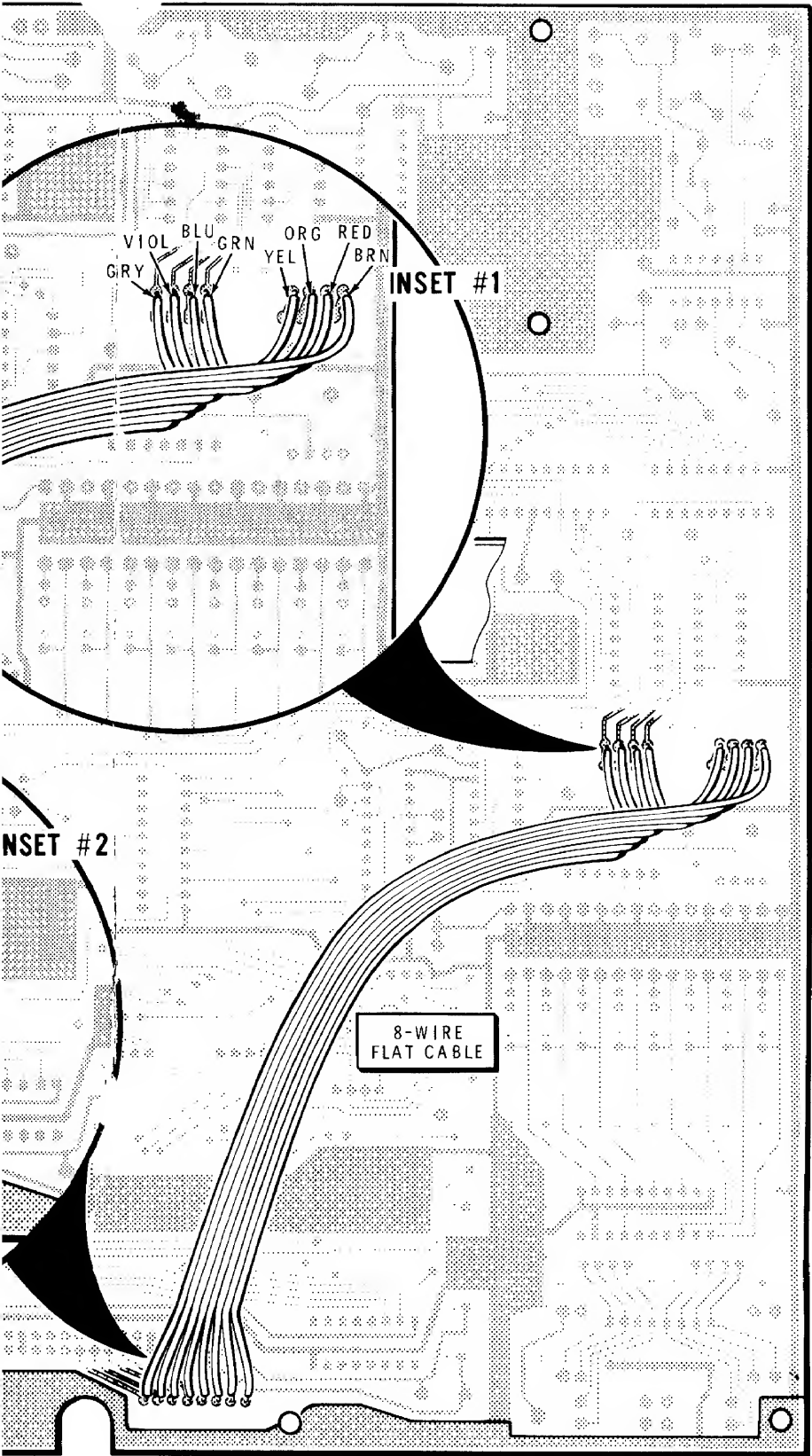
PICTORIAL 1-7



PARTIAL SCHEMATIC



PICTORIAL 2-4

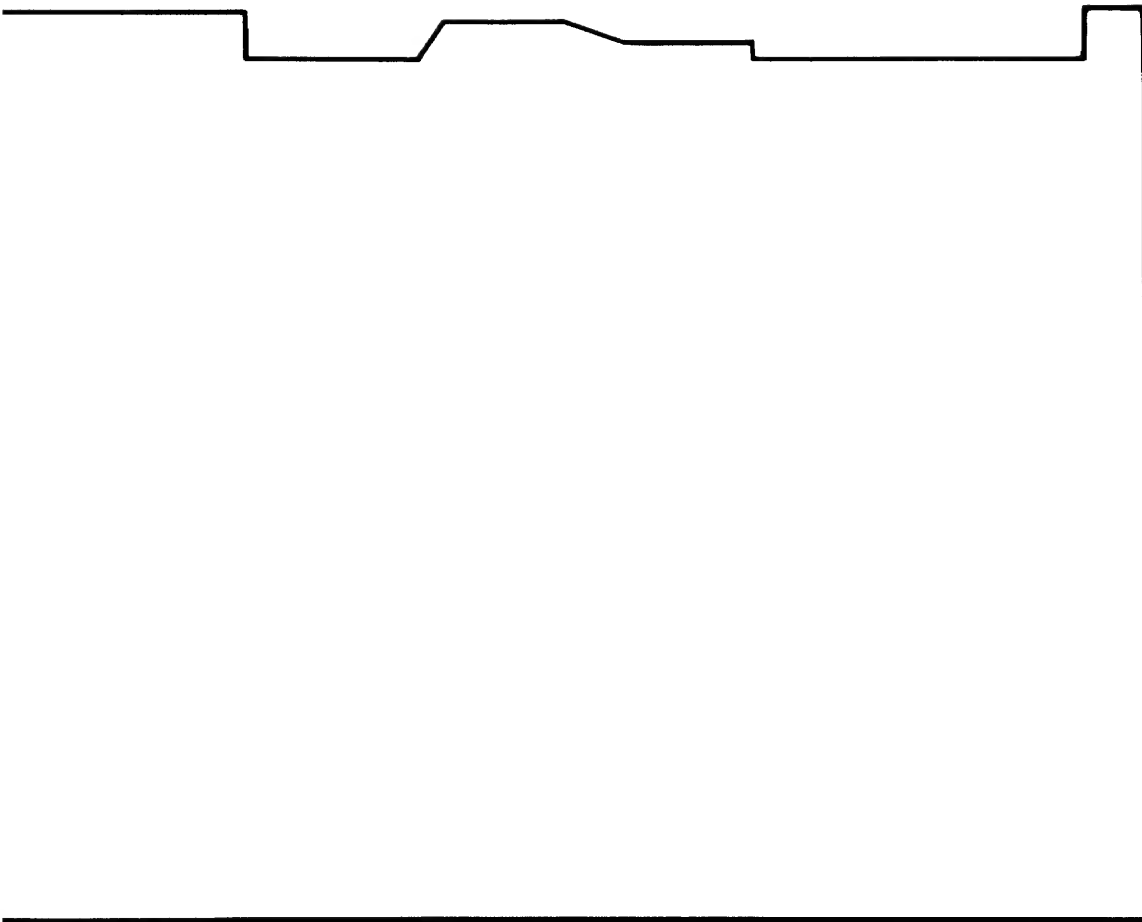


RF SHIELD
TEMPLATE

DULL SIDE
(FACE DOWN)

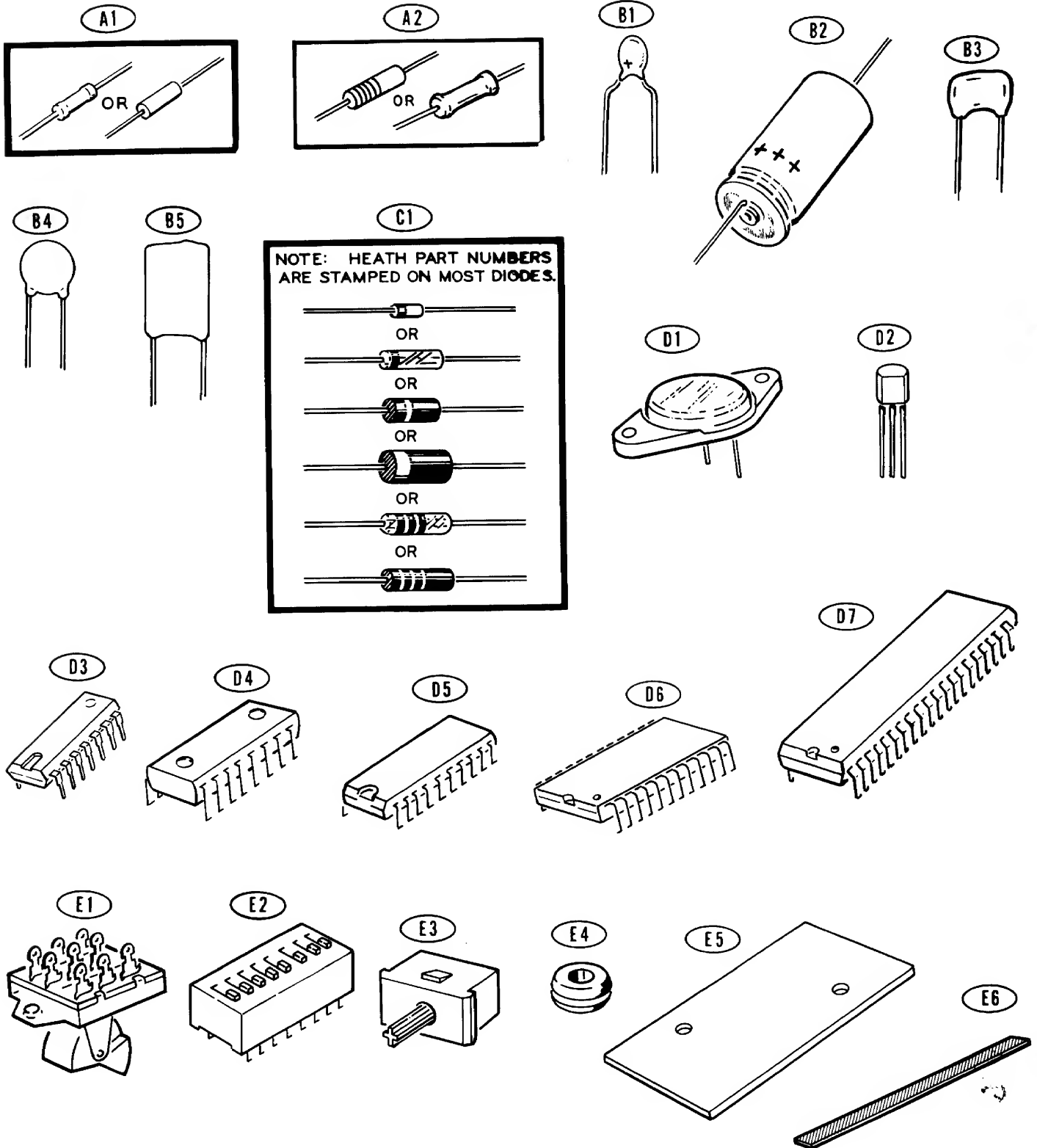


RF SHIELD
TEMPLATE



ILLUSTRATION

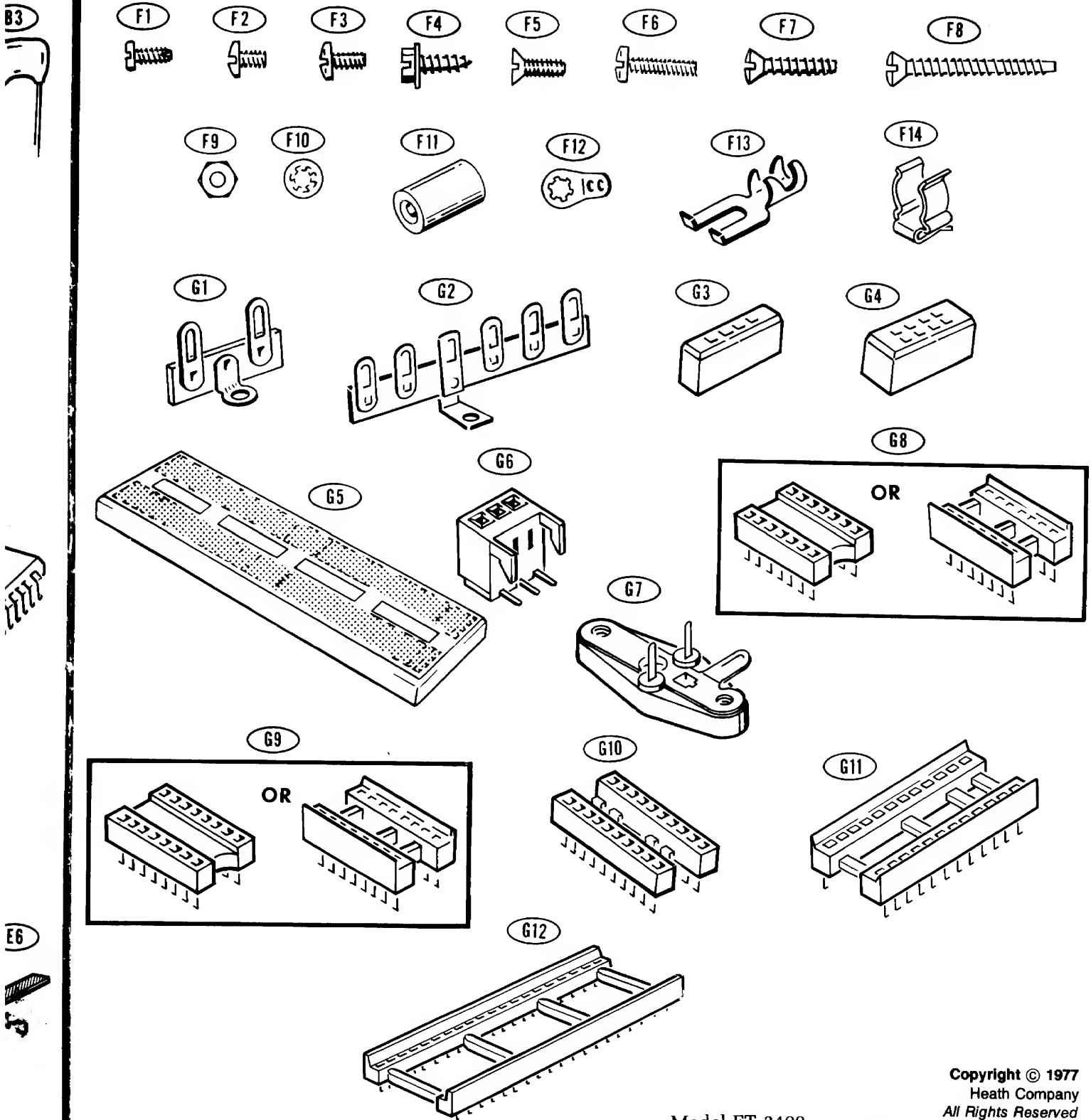
PARTS PICT



ATION BOOKLET

TS PICTORIAL

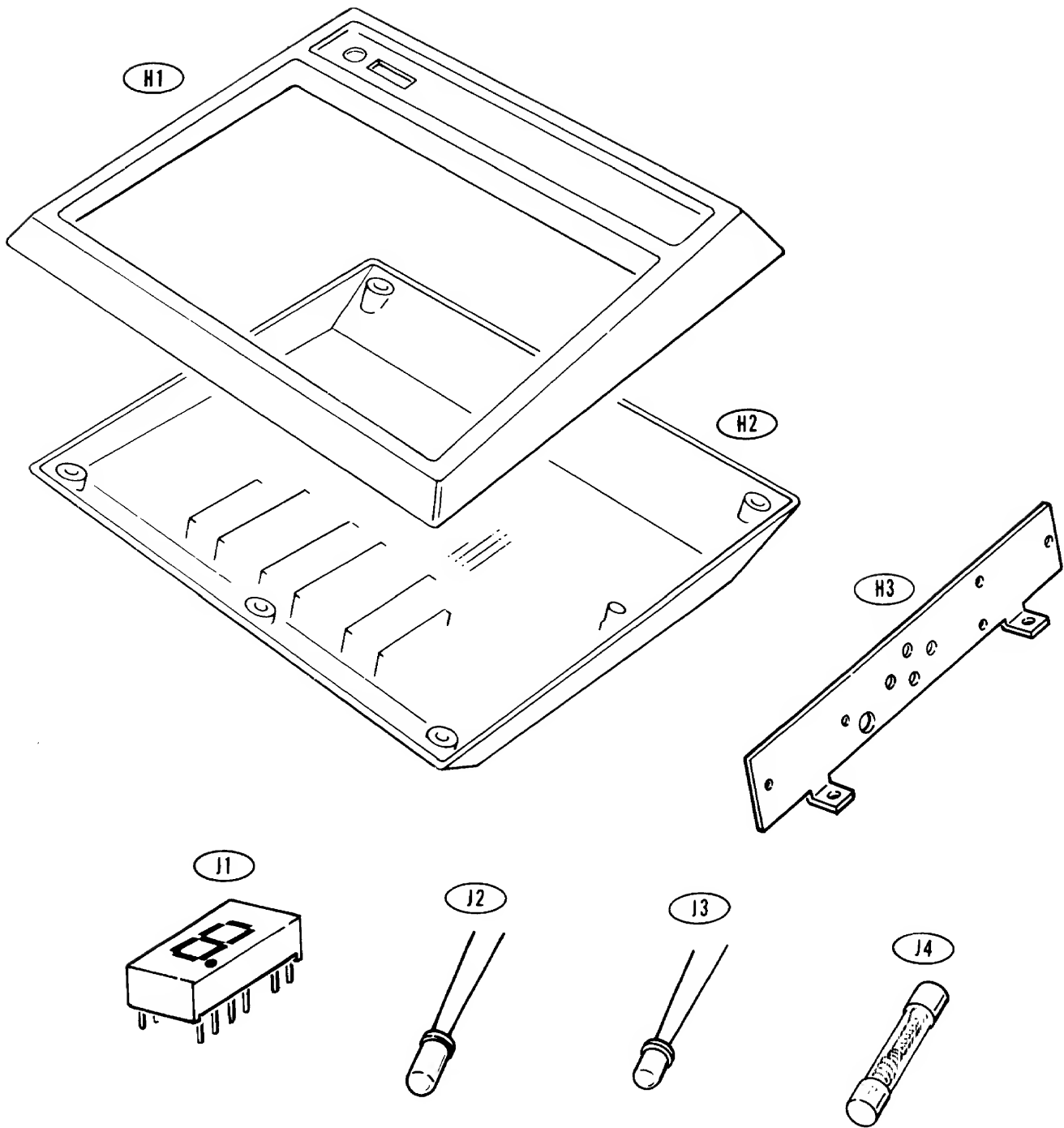
Part of 595-2021-06



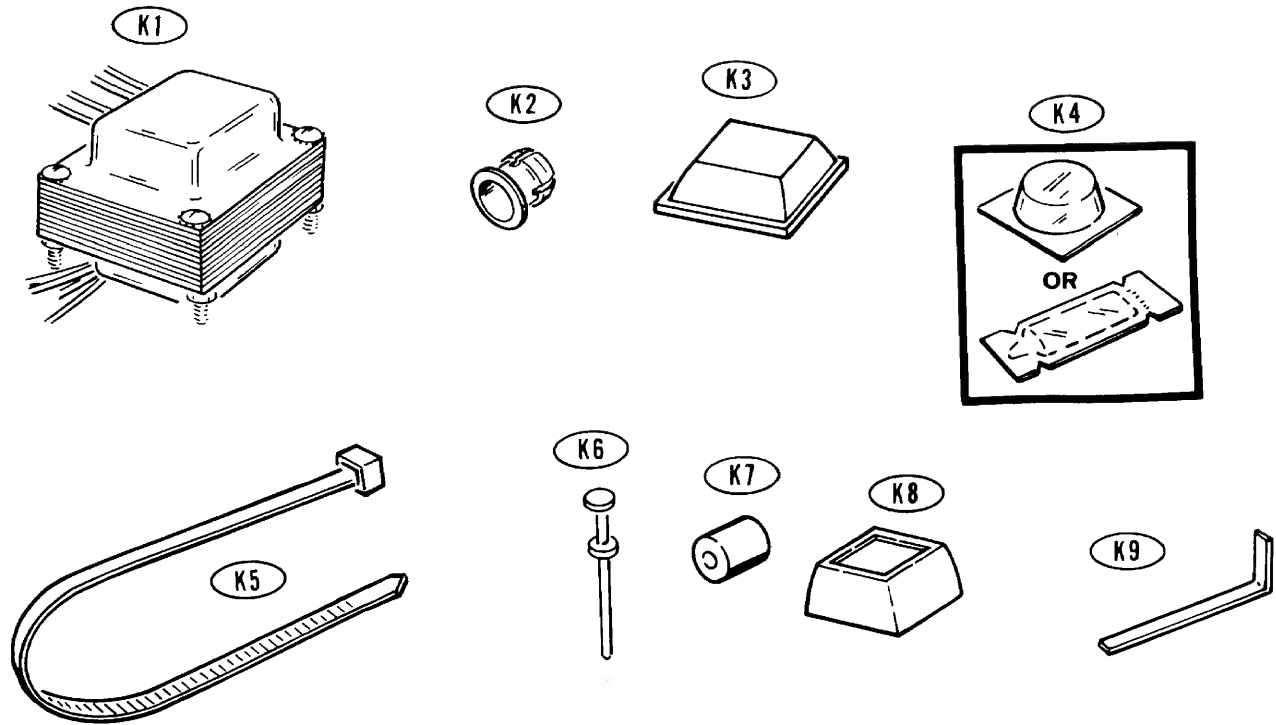
Model ET-3400

Copyright © 1977
Heath Company
All Rights Reserved
Printed in the United States of America

Parts Pictorial (c



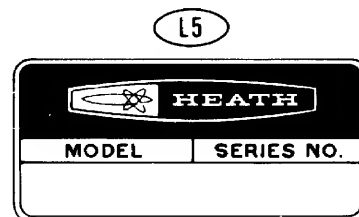
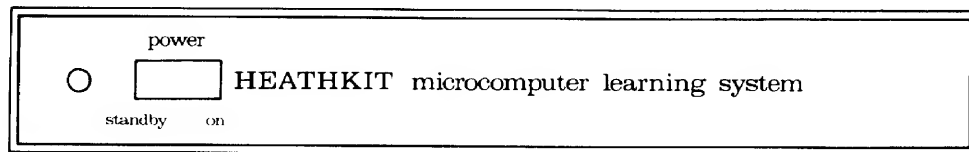
Pictorial (cont'd.)

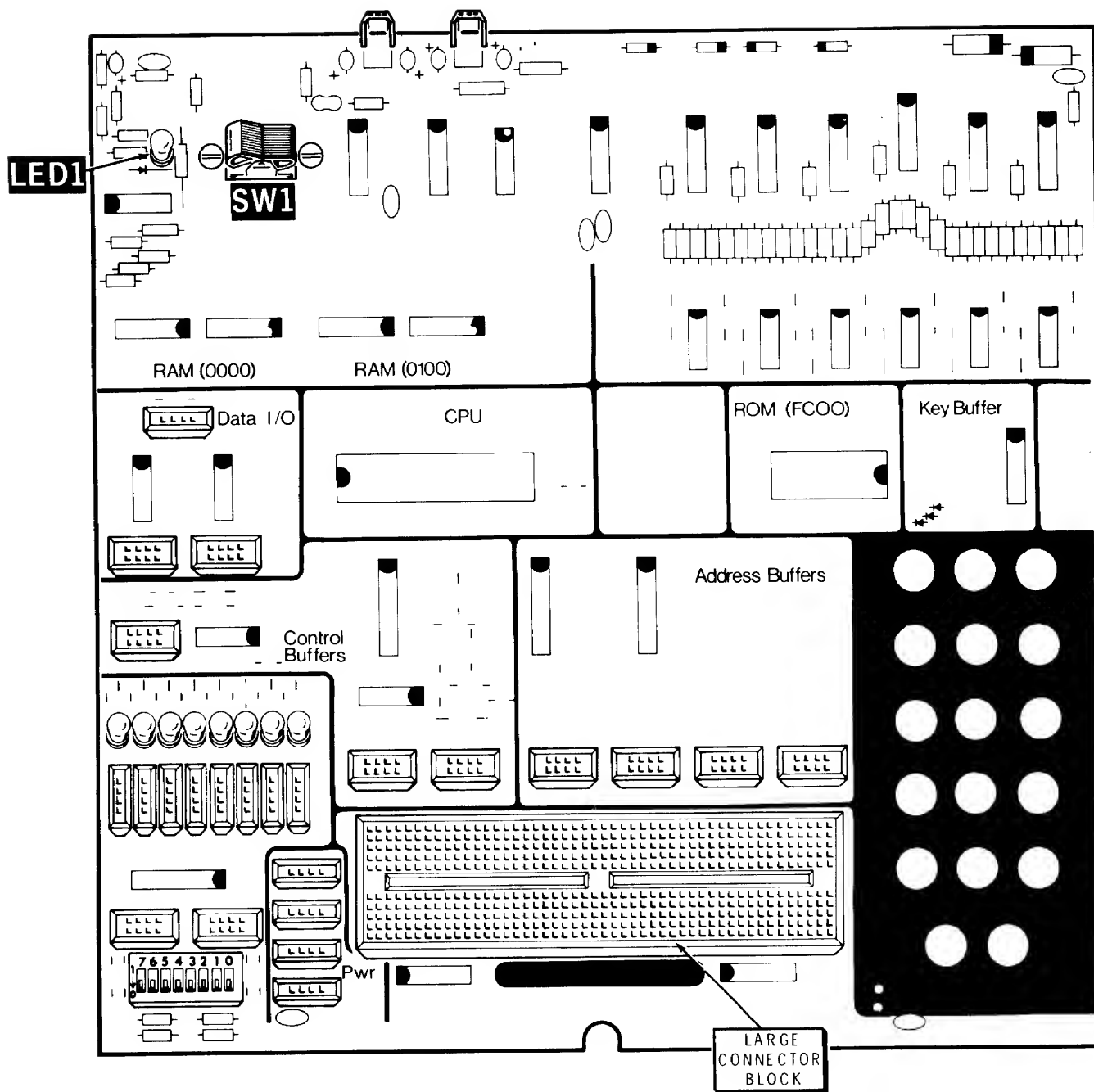


CAUTION: FOR CONTINUED PROTECTION
AGAINST FIRE HAZARD, REPLACE FUSE ONLY
WITH SAME TYPE AND RATING 390-1255

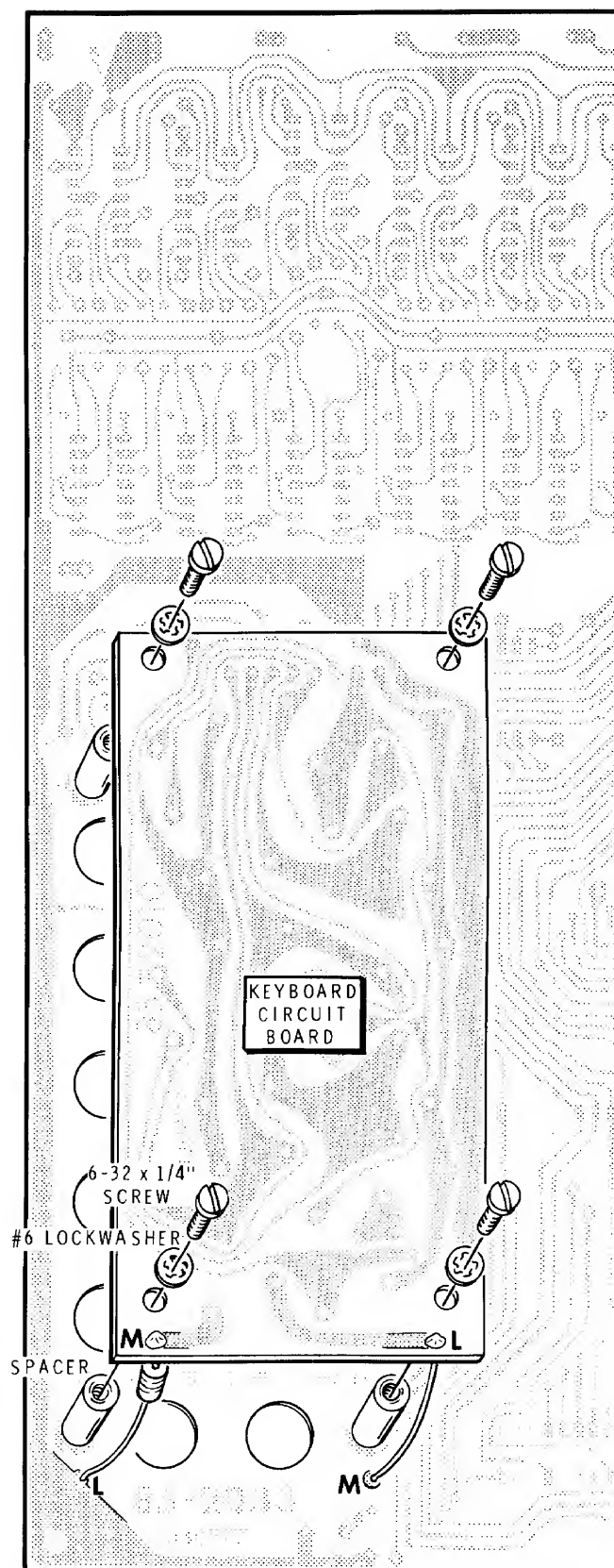
WARNING
DISCONNECT LINE CORD BEFORE OPENING

HEATH COMPANY
BENTON HARBOR, MICHIGAN 49022
MODEL ET 3400
120/240VAC 50/60Hz 30WATTS

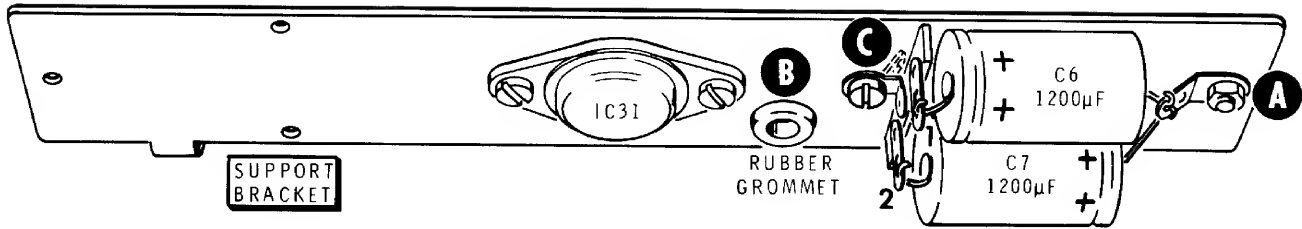




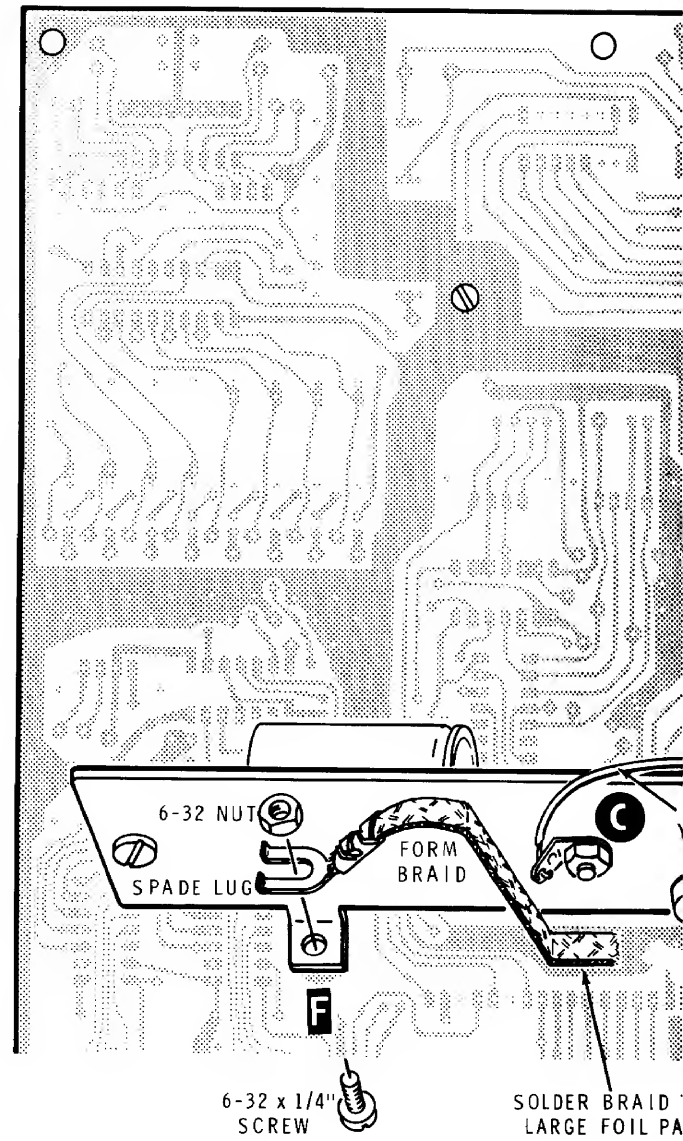
PICTORIAL 1-12

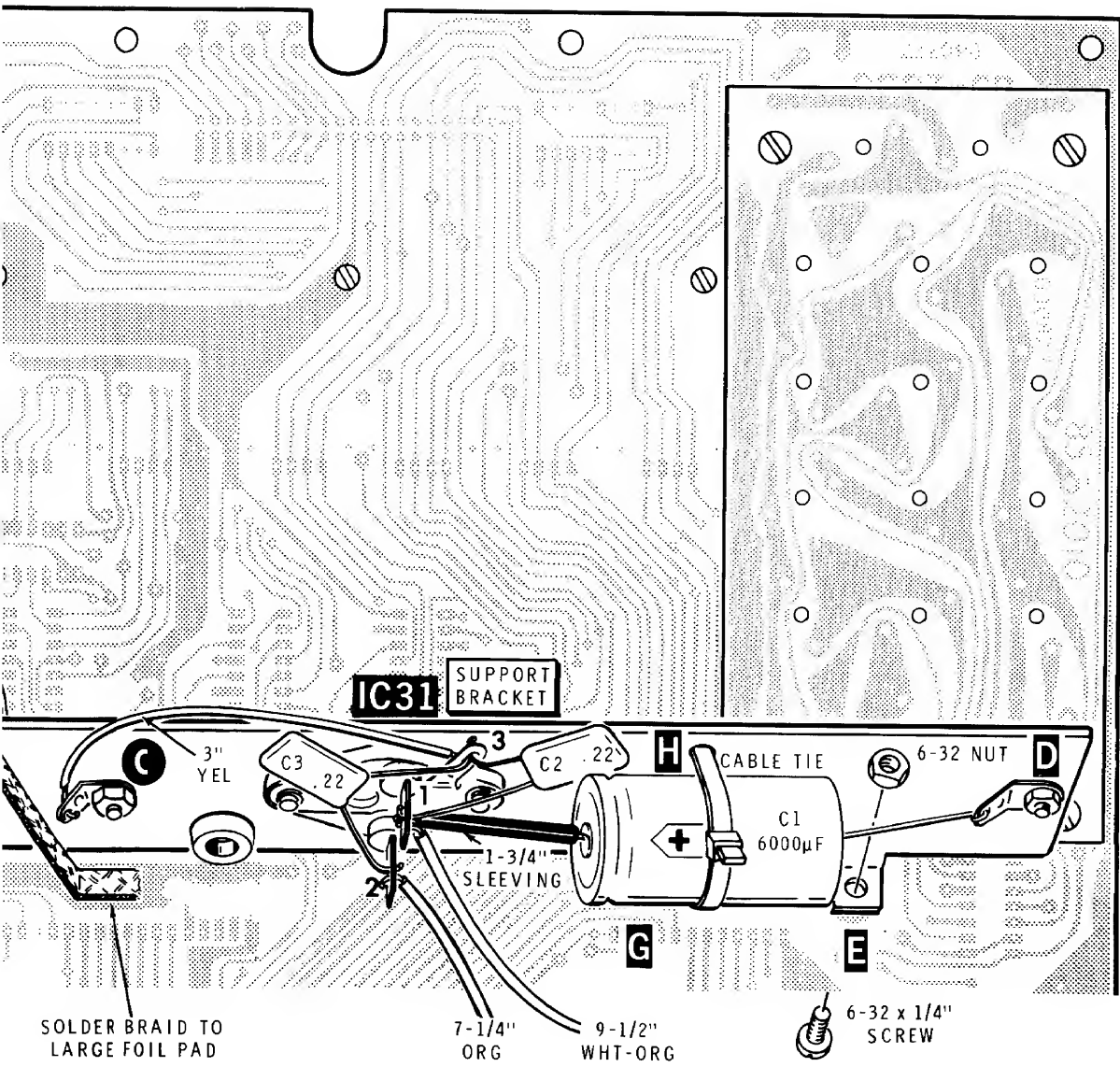


PICTORIAL 3-1

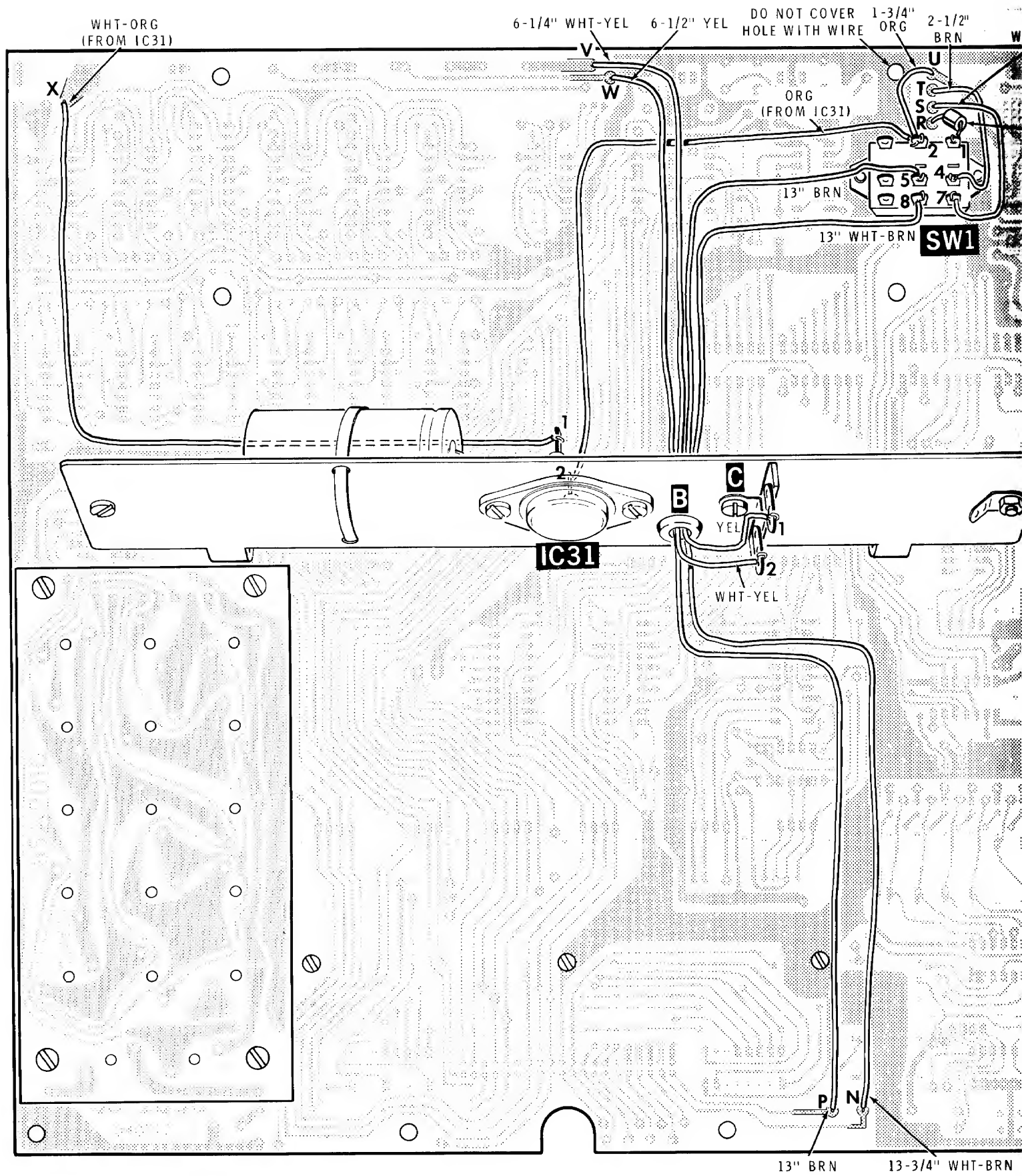


PICTORIAL 4-1

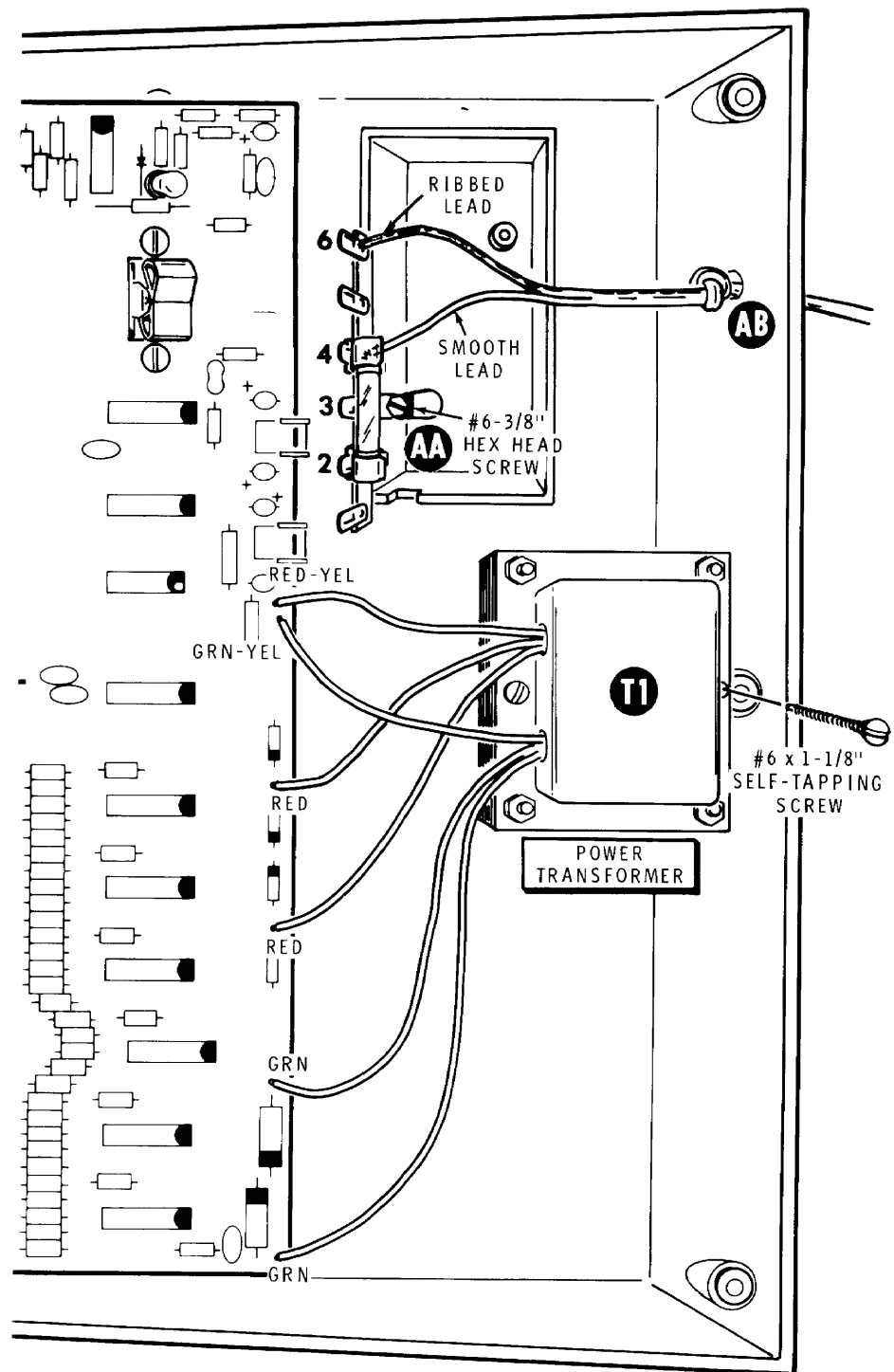
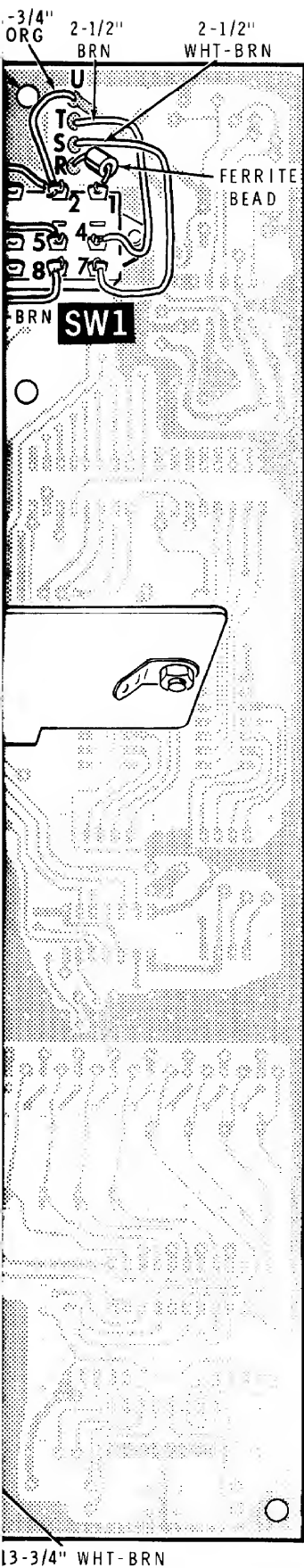




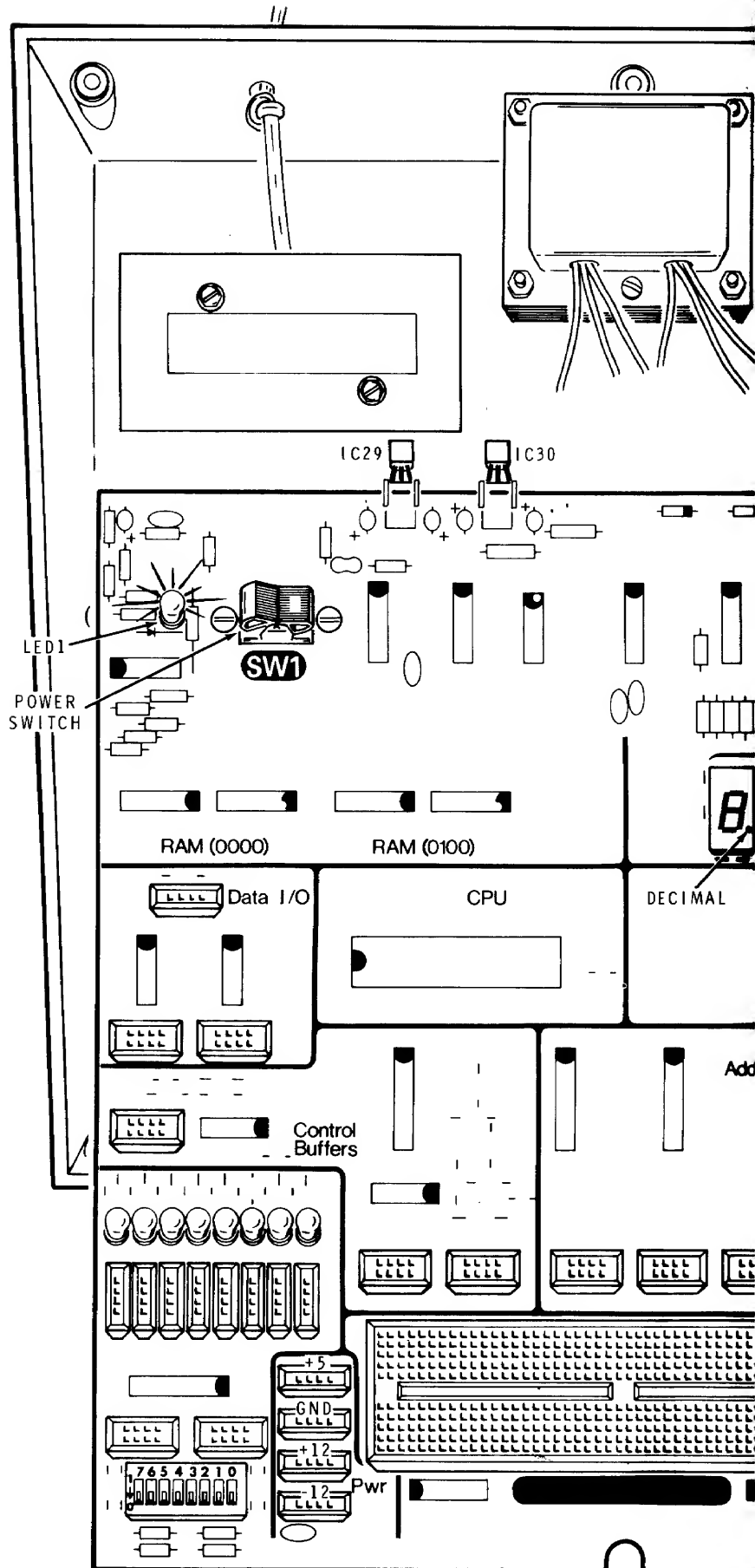
PICTORIAL 4-2

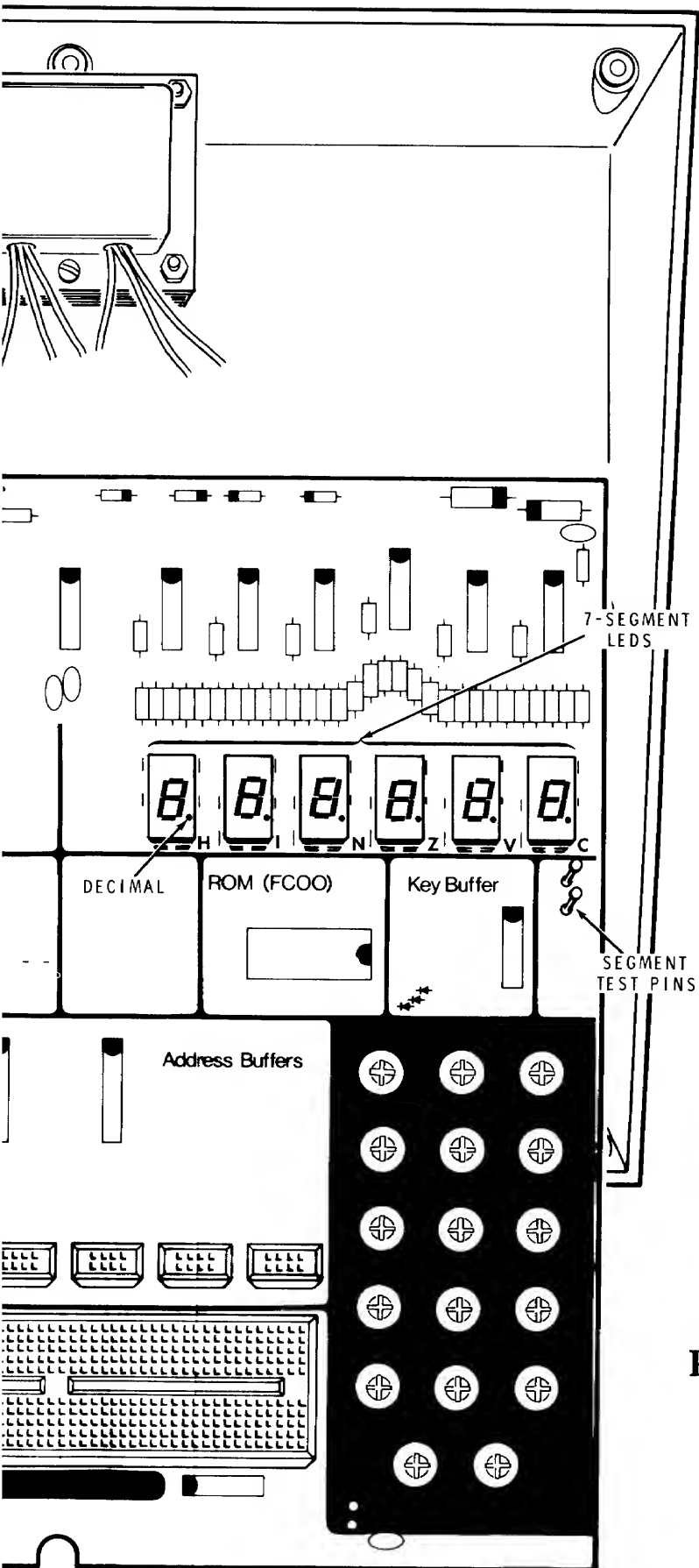


PICTORIAL 4-3

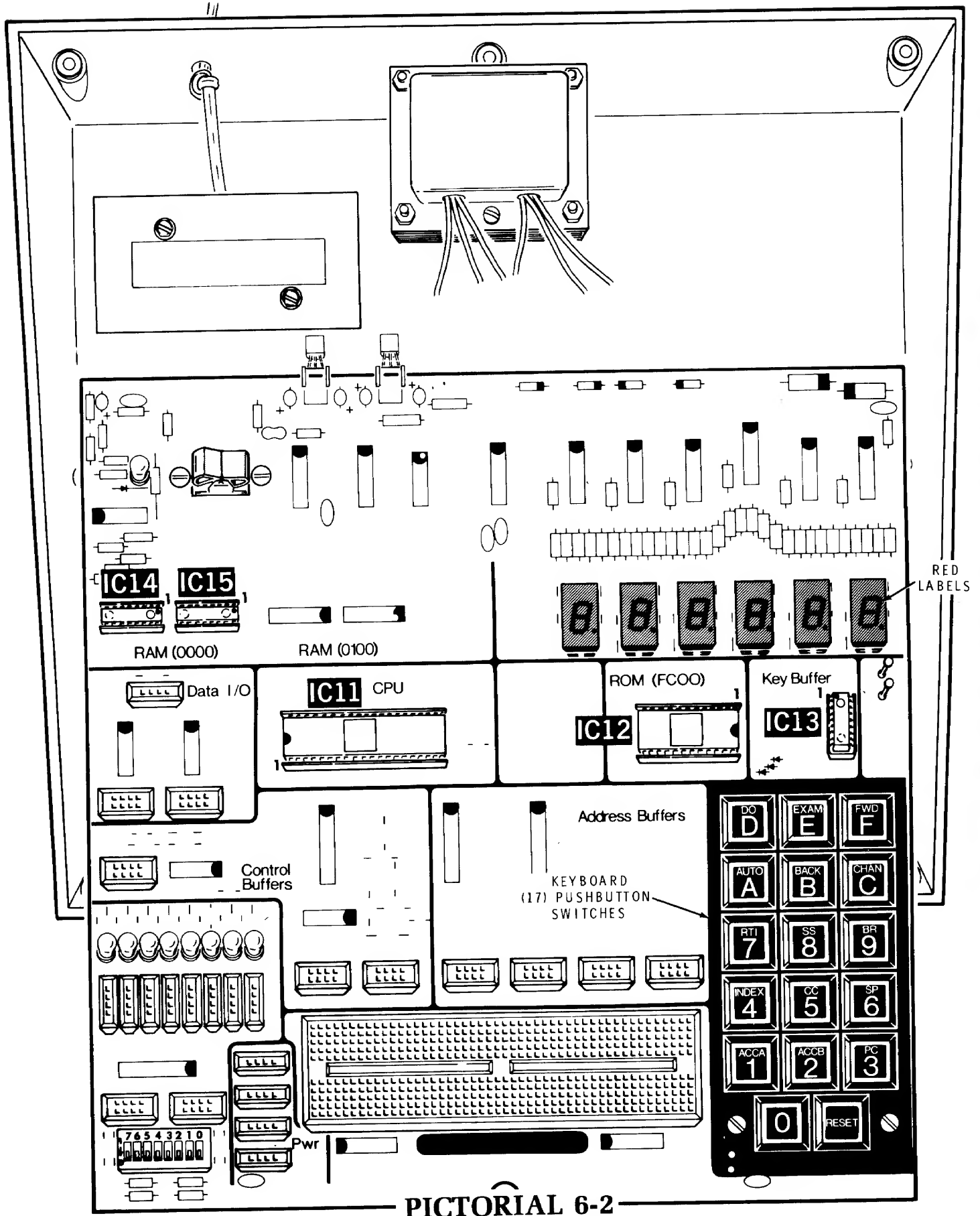


PICTORIAL 5-1

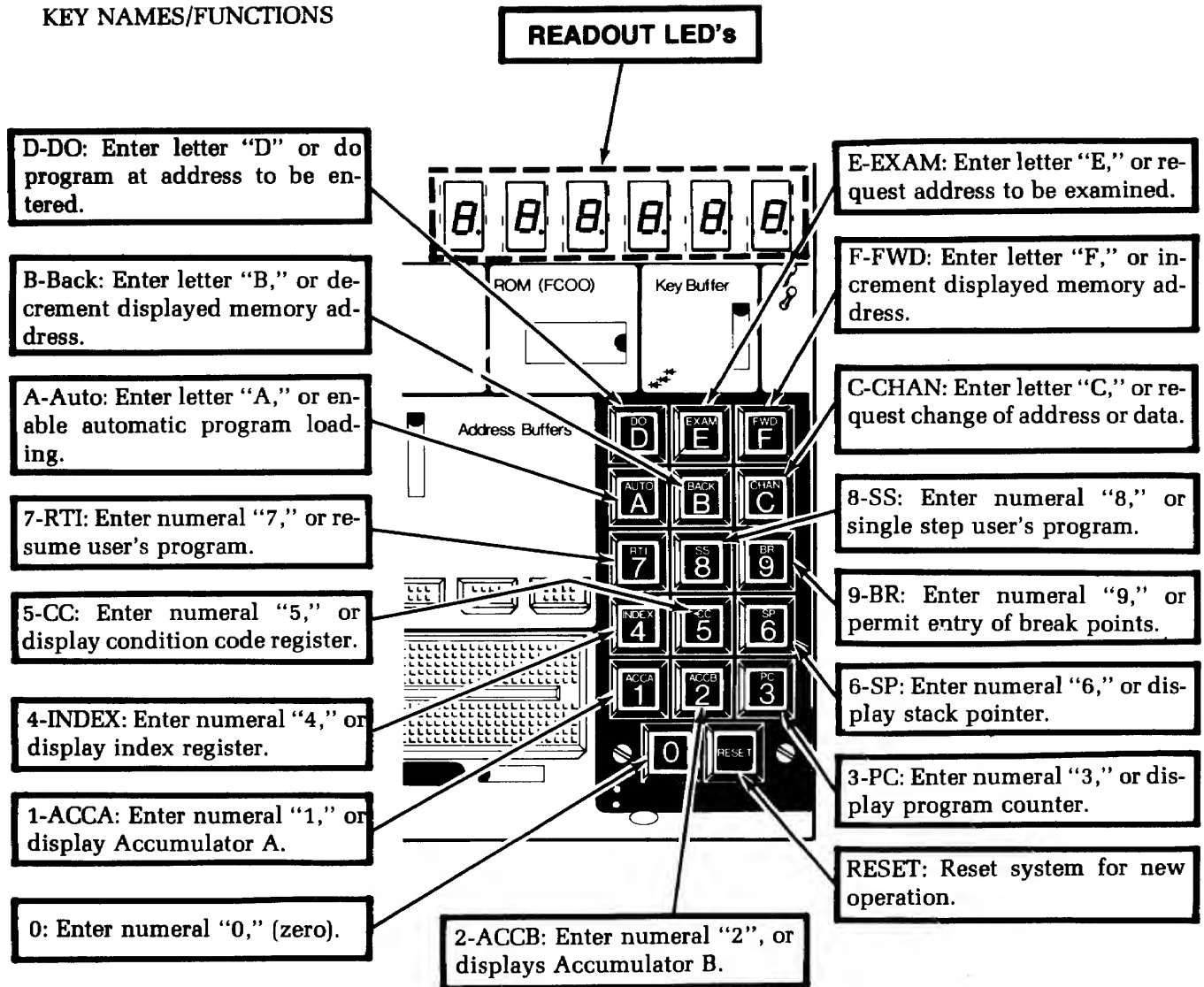




PICTORIAL 6-1



KEY NAMES/FUNCTIONS



PICTORIAL 6-3

POWER switch (SW1) - Selects either the STANDBY or ON position. Memory never dumps while the line cord is connected to AC power.

LED1 - Indicates when the line cord is plugged into AC power.

Provides input and output control of data transceivers.

Provides connections to system data lines.

Provides outputs from 02 clock, VMA02 clock, line, and 1Hz square wave sources.

LED's Display status of logic inputs.

Provides inputs for LED's.

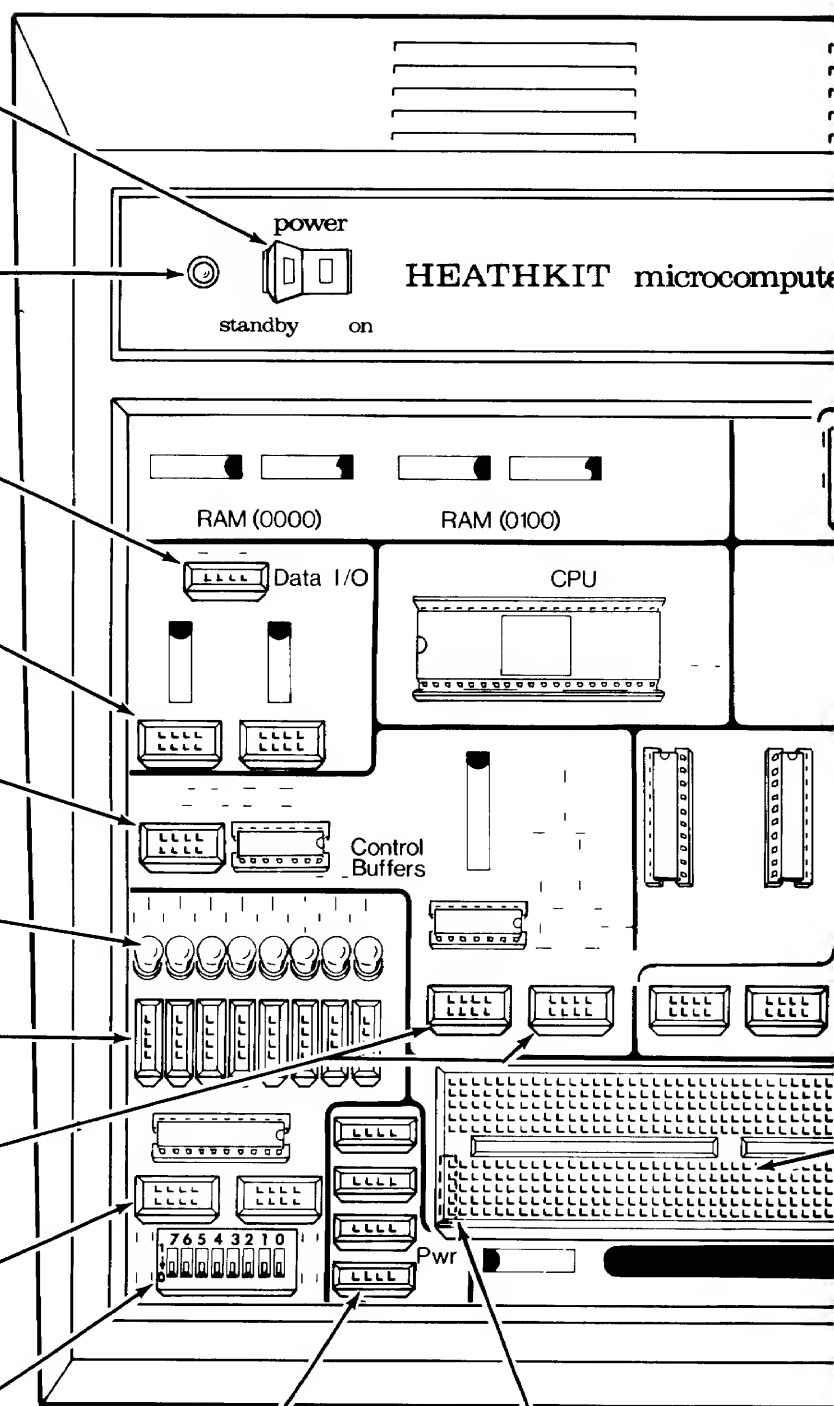
Provides connections to micro-processor control lines.

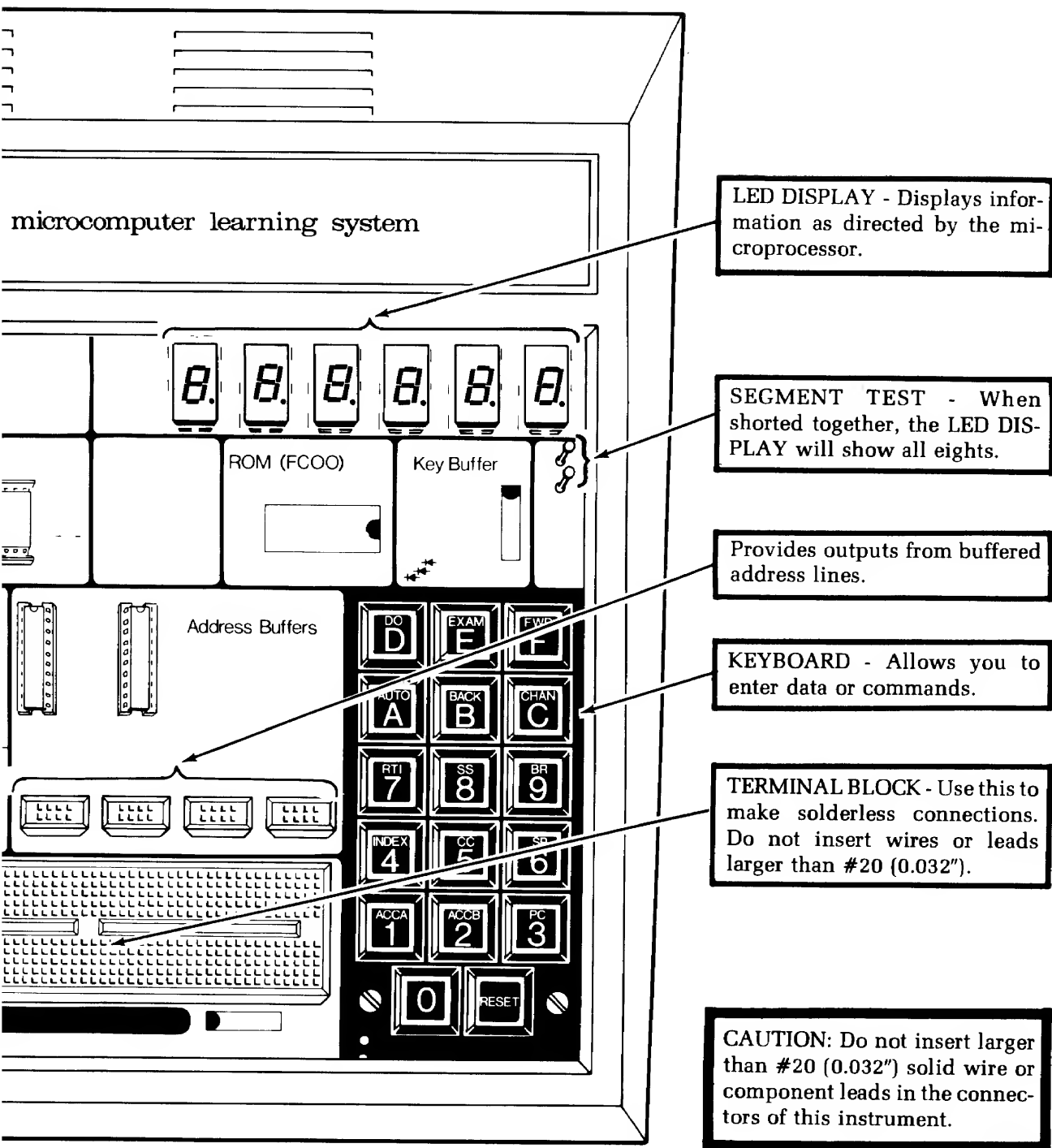
Provides outputs for INPUT SWITCHES.

INPUT SWITCHES - Provide logic 1's and 0's to 8-pin connectors.

Supplies connections to +12 volts, -12 volts, +5 volts, and ground.

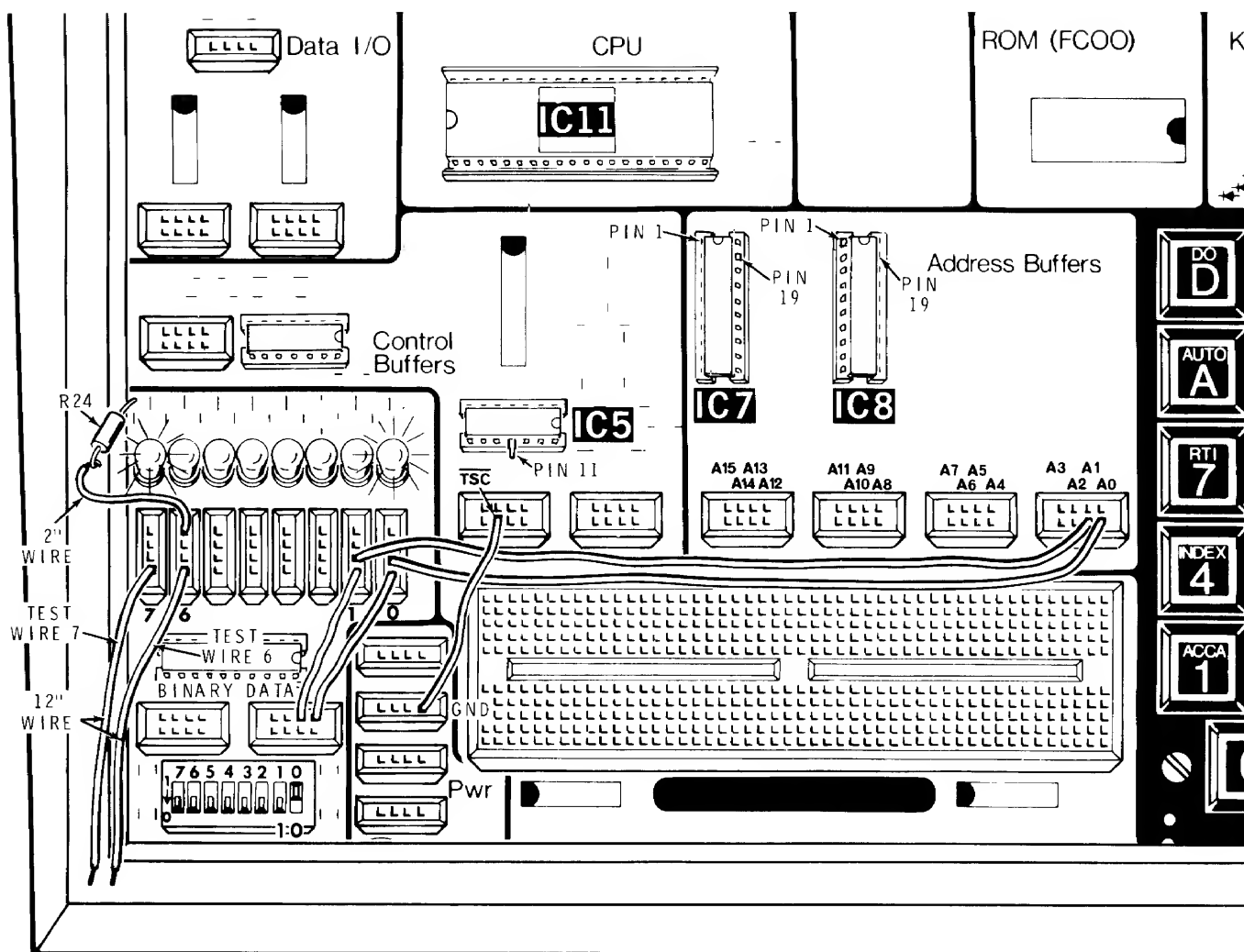
Connectors are internally connected together.



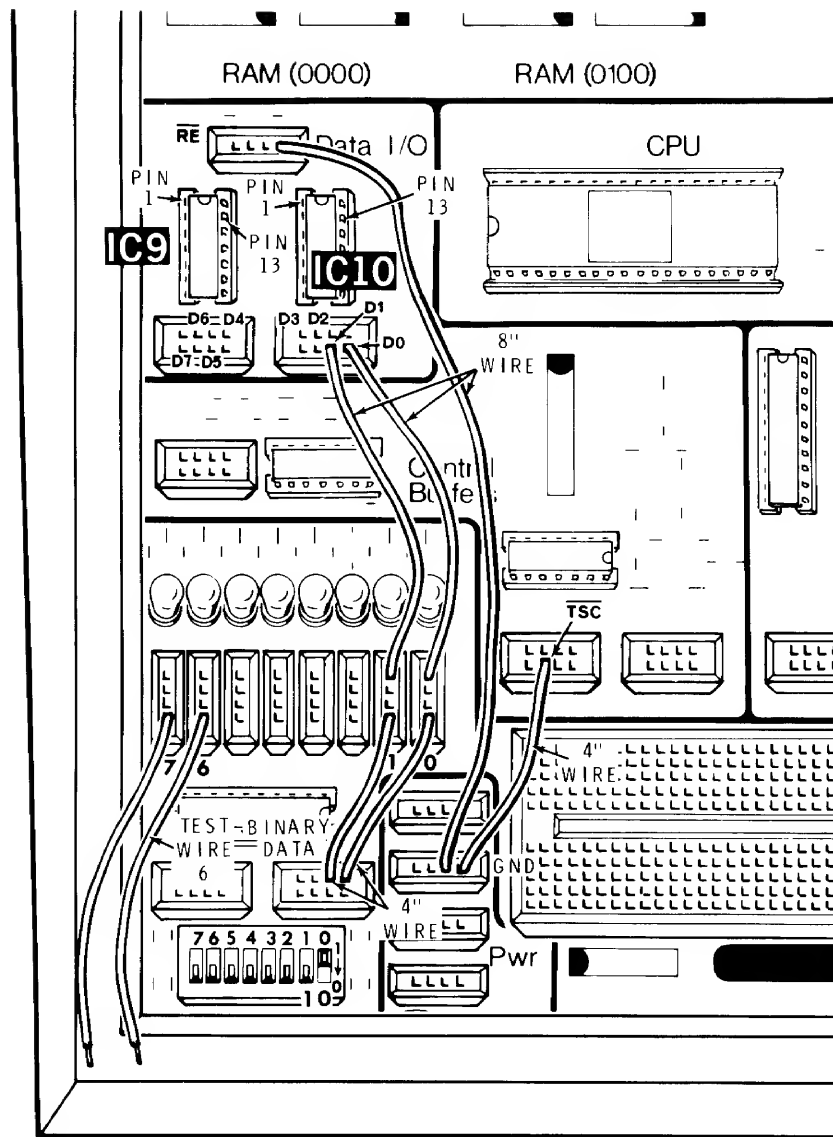


re internally con-
er.

PICTORIAL 8-1



PICTORIAL 9-5



PICTORIAL 9-6

MAIN DECODING C

	IC2							IC3							
	INPUT				OUTPUT			INPUT				OUTPUT		1, 2	3, 4
	12	13	14	15	1	7	9	12	13	14	15	1	5		
RAM 00XX	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0
RAM 01XX	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0
FFXX ROM FFX	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0
KEYBOARD C0-X	0	1	1	0	1	0	1	0	1	0	0	1	0	0	1
DISPLAY LED's C1XX	0	1	1	0	1	0	1	0	1	0	0	1	0	0	1

KEYBOARD COLUMN ADDRESS DEC

COLUMN ADDRESS	X				PRESS KEY	O
	A ₃	A ₂	A ₁	A ₀		
CO - 3	—	0	1	1	F C 9 6 3	
CO - 5	—	1	0	1	E B 8 5 2	
CO - 6	—	1	1	0	D A 7 4 1 0	

0 = LOGIC 0

1 = LOGIC 1

— = DOES NOT CARE

X = FUNCTIONING ADDRESS

ADDRESS DECODING CHART

	IC21					IC20								
PUT	1,	3,	12,	11,		INPUT				OUTPUT				END RESULT
5	2	4	13	5	6	12	13	14	15	5	6	10	11	
1	1	0	1	0	1	0	1	0	0	0	1	1	1	IC14 and IC15, Pin 13 is 0.
1	1	0	1	0	1	0	1	0	1	1	0	1	1	IC16 and IC17, Pin 13 is 0.
1	1	0	1	0	1	1	1	0	0	1	1	1	1	IC12 pins 10, 13, and 14 are 1. Pin 11 is 0.
0	0	1	0	1	0	1	0	0	0	1	1	0	1	IC13 pins 1 and 15 are 0.
0	0	1	0	1	0	1	0	0	1	1	1	1	0	IC22 pin.12 is 0.

ADDRESS DECODING CHART

PRESS KEY	LOGIC 0 ON DATA LINE
F	D ₀
C	D ₁
9	D ₂
6	D ₃
3	D ₄
E	D ₀
B	D ₁
8	D ₂
5	D ₃
2	D ₄
D	D ₀
A	D ₁
7	D ₂
4	D ₃
1	D ₄
0	D ₅

DISPLAY LED C1

	X				IC22					Logic output
					Input Pin					
	A ₇	A ₆	A ₅	A ₄	12	13	14	15		
LED C1	—	1	1	0	0	1	1	0	7	
		1	0	1	0	1	0	1	6	
	—	1	0	0	0	1	0	0	5	
	—	0	1	1	0	0	1	1	4	
	—	0	1	0	0	0	1	0	3	
	—	0	0	1	0	0	0	1	2	

	X				IC23 through IC28, input pins 3, 2, 1			IC23 through IC28, Output
	A ₃	A ₂	A ₁	A ₀				
LED segment C1X	1	1	1	0	1	1	0	11
	1	1	0	1	1	0	1	10
	1	1	0	0	1	0	0	9
	1	0	1	1	0	1	1	8
	1	0	1	0	0	1	0	7
	1	0	0	1	0	0	1	6
	1	0	0	0	0	0	0	5
	1	1	1	1	1	1	1	12

*With a given output pin addressed, the logic level on that pin is the logic level on the addressed IC.

D₀ LOGIC LEVEL CHARACTERISTICS

D ₀ logic levels	D ₀	IC21		IC6
		Pins 9 and 10	Pin 2	
	1	1	0	0

DISPLAY LED CHART

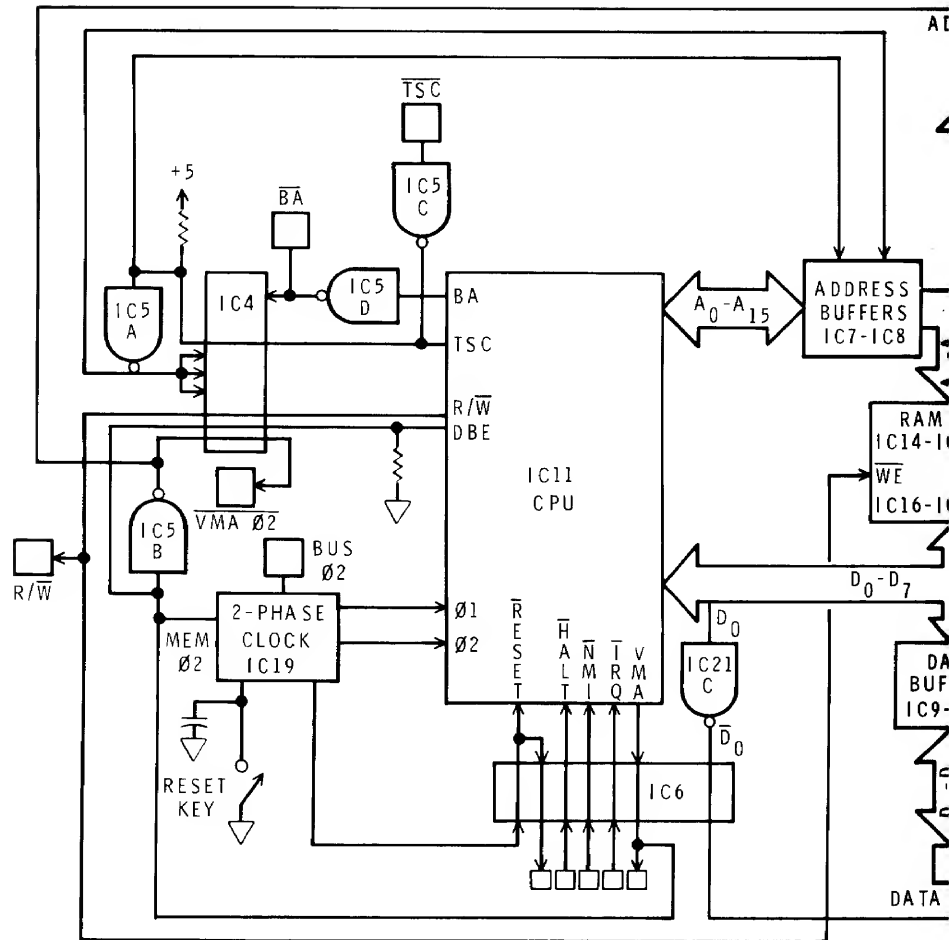
IC22		Logic 0 at output pin	Logic 0 on enable pin 14 of IC	Display LED addressed
Pin 14	15			
1	0	7	23	H
0	1	6	24	I
0	0	5	25	N
1	1	4	26	Z
1	0	3	27	V
0	1	2	28	C

through input 2, 1	IC23 through IC28* Output pin	LED pin	Segment*
1 0	11	1	a
0 1	10	13	b
0 0	9	10	c
1 1	7	8	d
1 0	6	7	e
0 1	5	2	f
0 0	4	11	g
1 1	12	9	DP

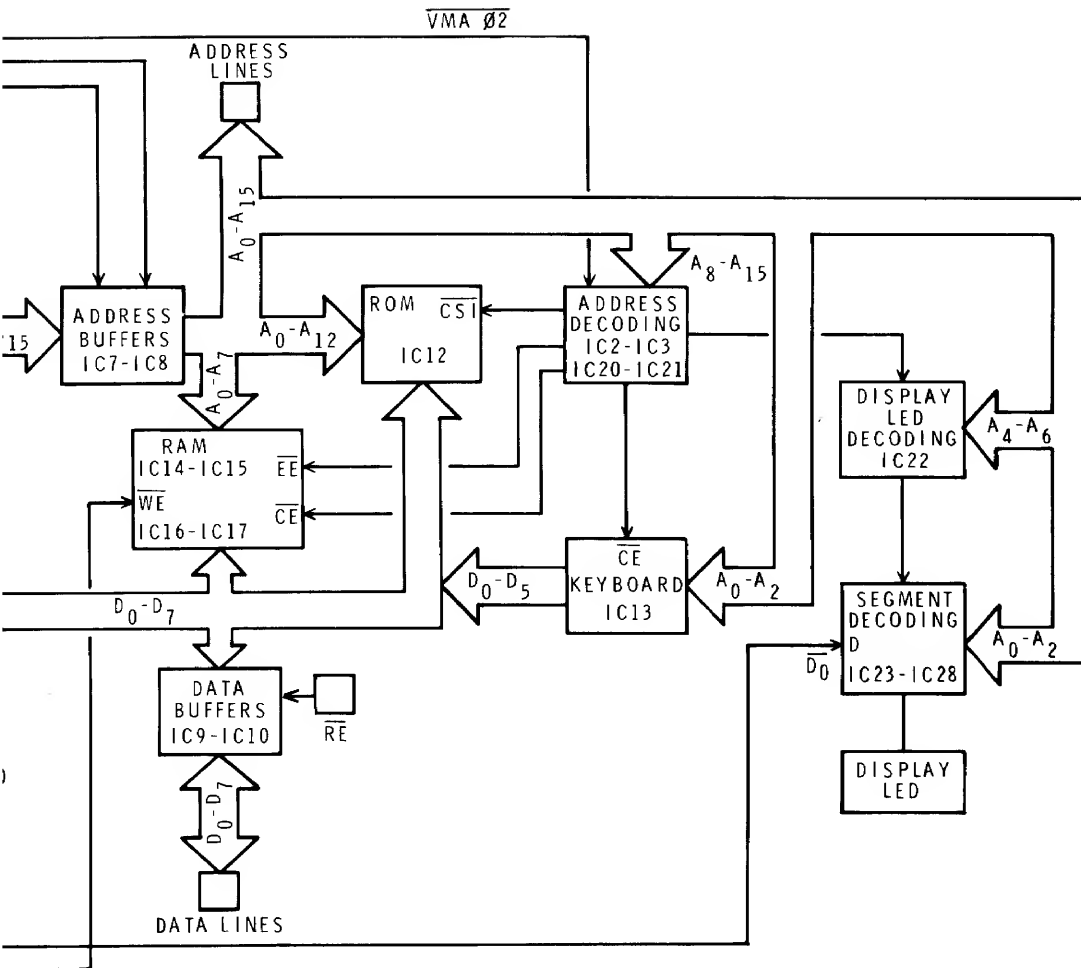
logic level on that pin will follow the level on the D input of the

IC LEVEL CHART

IC6		IC23 through IC28 Pin 13
Pin 2	Pin 18	
0	0	0



BLOCK DIAG



OCK DIAGRAM

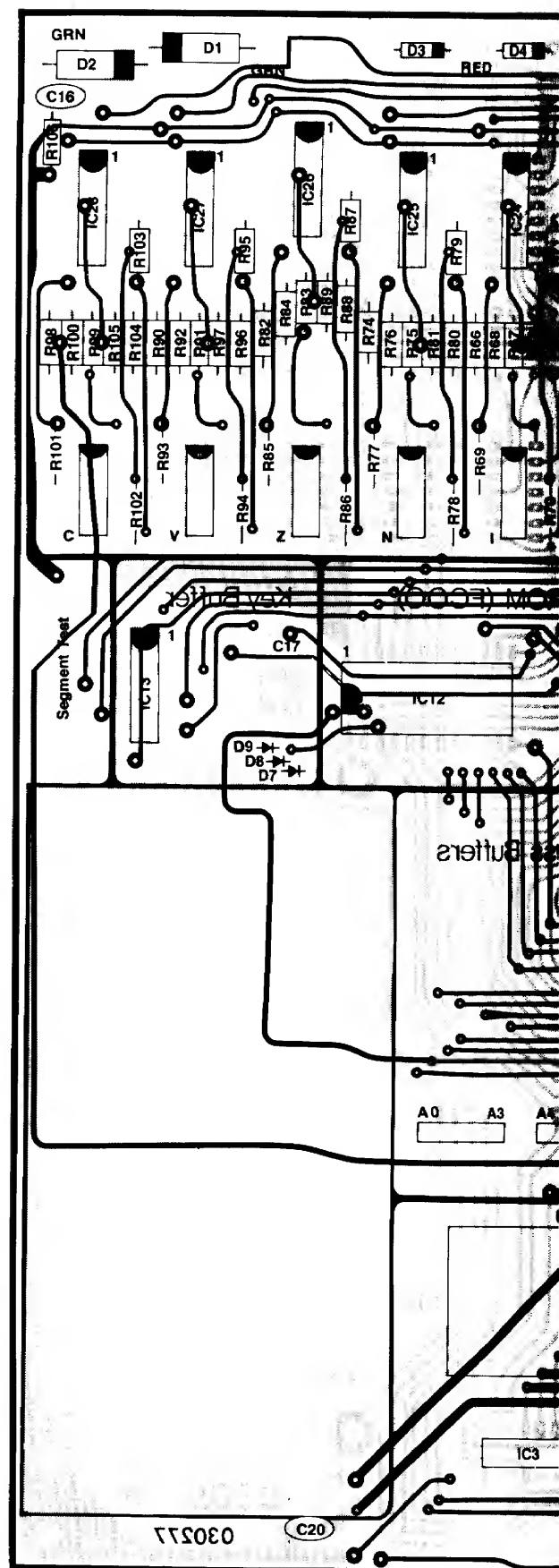
CIRCUIT BOARD X-RAY VIEW

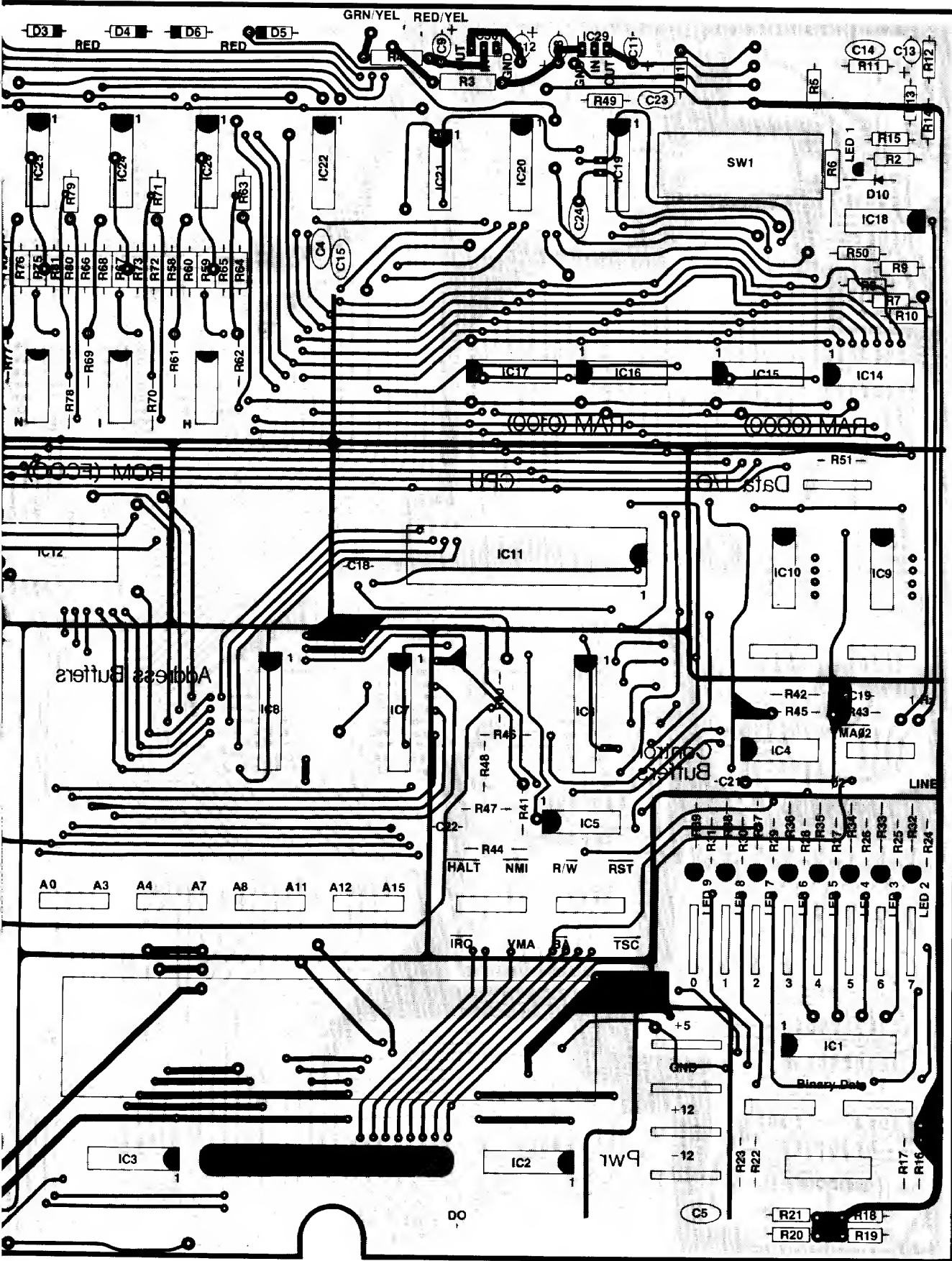
NOTE: To find the PART NUMBER of a component for the purpose of ordering a replacement part:

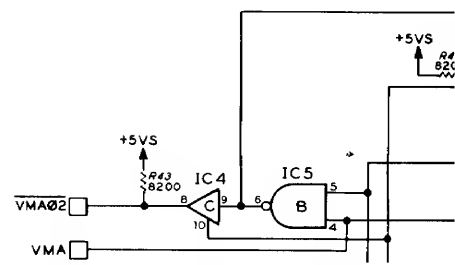
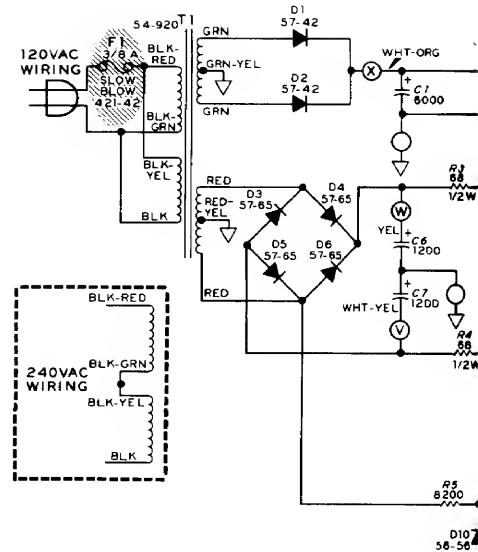
- A. Find the circuit component number (R5, C3, etc.) on the "X-Ray View."
- B. Locate this same number in the "Circuit Component Number" column of the "Parts List."
- C. Adjacent to the circuit component number, you will find the PART NUMBER and DESCRIPTION which must be supplied when you order a replacement part.

Top foil in red

(Shown from bottom side)





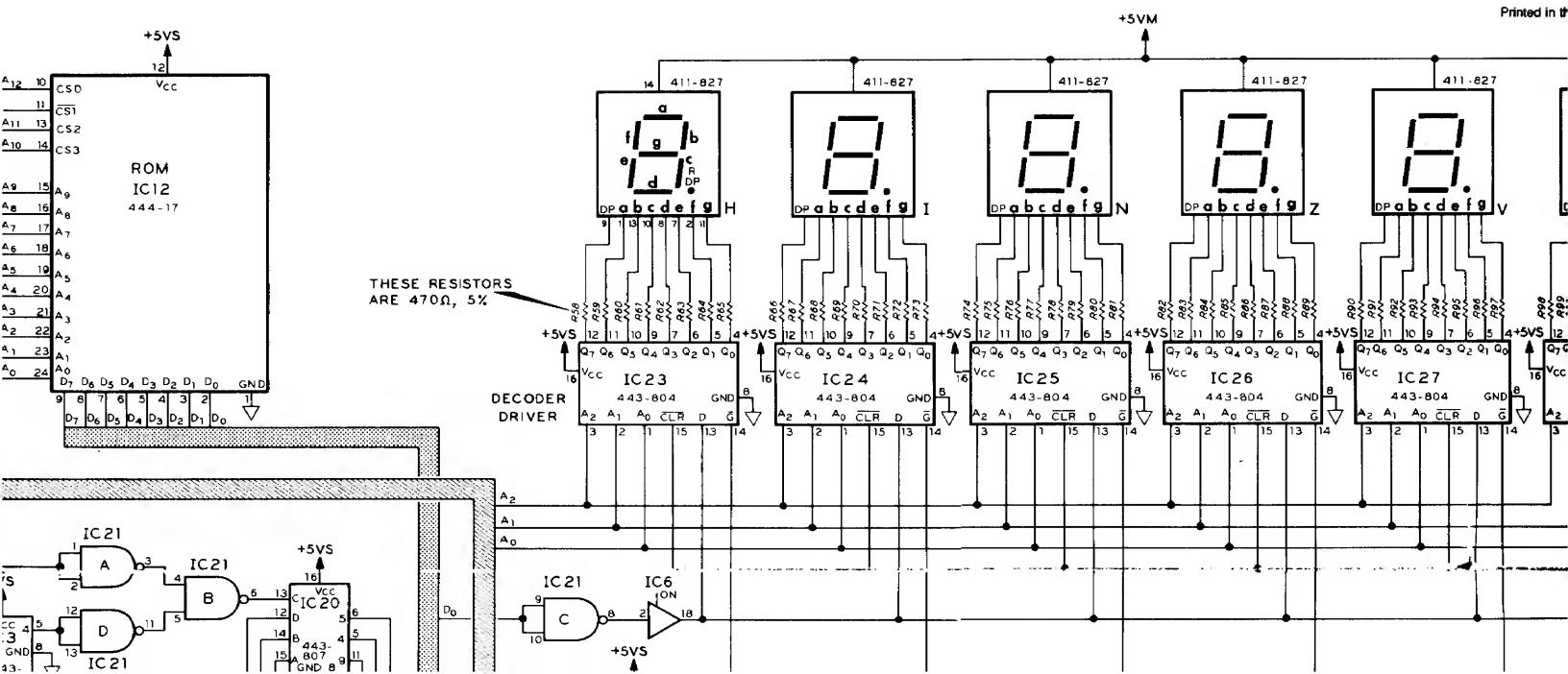
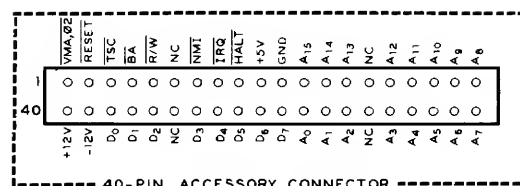


5;
 ALL RESISTORS ARE 1/4 WATT, 10% UNLESS MARKED OTHERWISE. RESISTOR VALUES ARE IN OHMS (k=1000; M=1,000,000).
 ALL CAPACITORS ARE IN μ F UNLESS MARKED OTHERWISE.
 THIS SYMBOL INDICATES A DC VOLTAGE TAKEN WITH A HIGH IMPEDANCE INPUT VOLTMETER FROM THE POINT INDICATED TO CHASSIS GROUND. VOLTAGES MAY VARY $\pm 20\%$.
 THIS SYMBOL INDICATES CIRCUIT BOARD GROUND.
 THIS SYMBOL INDICATES A CONNECTOR IN A CONNECTOR BLOCK.

USE IS CRITICAL FOR CONTINUED SAFETY. REPLACE THEM ONLY WITH PARTS OF THE SAME RATING OR WITH THE PROPER HEATH PARTS.

IF YOU INSTALL THE 40-PIN ACCESSORY CONNECTOR, CONNECTIONS ARE TO THE BUFFERED LINES (\square) USE THUMPER WIRES TO CONNECT THE DATA LINES.

SCHEMATIC OF THE HEATHKIT[®] MICROCOMPUTER LEARNING SYSTEM MODEL ET-3400



Part of 595-2021-06

Copyright © 1977

Heath Company

All Rights Reserved

Printed in the United States of America

